Robust Clustering Using Outlier-Sparsity Regularization

Pedro A. Forero, *Student Member, IEEE,* Vassilis Kekatos, *Member, IEEE,* and Georgios B. Giannakis (Contact Author)*, *Fellow, IEEE*

Abstract

Notwithstanding the popularity of conventional clustering algorithms such as K-means and probabilistic clustering, their clustering results are sensitive to the presence of outliers in the data. Even a few outliers can compromise the ability of these algorithms to identify meaningful hidden structures rendering their outcome unreliable. This paper develops robust clustering algorithms that not only aim to cluster the data, but also to identify the outliers. The novel approaches rely on the infrequent presence of outliers in the data which translates to sparsity in a judiciously chosen domain. Capitalizing on the sparsity in the outlier domain, outlier-aware robust K-means and probabilistic clustering approaches are proposed. Their novelty lies on identifying outliers while effecting sparsity in the outlier domain through carefully chosen regularization. A block coordinate descent approach is developed to obtain iterative algorithms with convergence guarantees and small excess computational complexity with respect to their non-robust counterparts. Kernelized versions of the robust clustering algorithms are also developed to efficiently handle high-dimensional data, identify nonlinearly separable clusters, or even cluster objects that are not represented by vectors. Numerical tests on both synthetic and real datasets validate the performance and applicability of the novel algorithms.

Index Terms

(Block) coordinate descent, clustering, expectation-maximization algorithm, Group-Lasso, kernel methods, K-means, mixture models, robustness, sparsity.

Part of this work was presented at the 36th IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, Prague, Czech Republic, May 2011. Work was in part supported by NSF grant CCF-1016605; Dr. Kekatos was funded by the European Community's Seventh Framework Programme (FP7/2008 under grant agreement No. 234914). The authors are with the ECE Dept., University of Minnesota, Minneapolis, MN 55455, USA, Emails:{forer002,kekatos,georgios}@umn.edu

I. Introduction

Clustering aims to partition a set of data into subsets, called clusters, such that data assigned to the same cluster are similar in some sense. Working with unlabeled data and under minimal assumptions makes clustering a challenging, yet universal tool for revealing data structures in a gamut of applications such as DNA microarray analysis and bioinformatics, (social) network analysis, image processing, and data mining [31], [14]. Moreover, clustering can serve as a pre-processing step for supervised learning algorithms in applications where labeling data one-at-a-time is costly. Multiple interpretations across disciplines of what a cluster is, have led to an abundant literature of application-specific algorithms [31].

Among the algorithms which cluster data represented by vectors, K-means and Gaussian mixture model (GMM-)based clustering are two popular schemes [22], [31]. Conventional K-means relies on the Euclidean distance as a similarity measure, thereby yielding partitions that minimize the within-cluster scatter [14]. Contrastingly, soft (a.k.a. fuzzy) K-means is tailored to identify overlapping clusters by allowing each datum to belong to multiple clusters [1]. GMM-based clustering considers observed data drawn from a probability density function (pdf) following a GMM, where each class-conditional pdf corresponds to a cluster [31]. Clustering arises as a by-product of a maximum likelihood (ML) estimation framework for the GMM parameters. ML parameter estimates are typically obtained through the expectation-maximization (EM) algorithm [9]. Kernel methods have been devised to enable clustering of nonlinearly separable clusters [26], [25].

Notwithstanding their popularity, K-means and GMM-based clustering are sensitive to inconsistent data, termed outliers, due to their functional dependency on the Euclidean distance [16]. Outliers appear infrequently in the data, emerging either due to reading errors or because they belong to rarely-seen and hence, markedly informative phenomena. However, even a few outliers can render clustering unreliable: cluster centers and model parameter estimates can be severely biased, and thus the data-to-cluster assignment is deteriorated. This motivates robustifying clustering approaches against outliers at affordable computational complexity in order to unravel the underlying structure in the data.

Robust clustering approaches to clustering have been investigated. In [8] and [15], an additional cluster intended for grouping outliers is introduced with is centroid assumed equidistant from all non-outlying data. Possibilistic clustering measures the so-called typicality of each datum with respect to (wrt) each cluster to decide whether a datum is an outlier [20], [24]. However, possibilistic clustering is sensitive to initialization and can output the same cluster more than once. Clustering approaches originating from robust statistics, such as the minimum volume ellipsoid and Huber's ϵ-contaminated model-based methods

[18], [34], extract one cluster at a time. This deflation approach can hinder the underlying data structure by removing elements before seeking other clusters. Other approaches rooted on robust statistics are based on the ℓ_1-distance (K-medians), Tukey's biweighted function, and trimmed means [3], [19], [12], [32]; but are all limited to linearly separable clusters.

The first contribution of the present work is to introduce a data model for clustering that explicitly accounts for outliers via a deterministic outlier vector per datum (Section II). A datum is deemed an outlier if its corresponding outlier vector is nonzero. Translating the fact that outliers are rare to *sparsity* in the outlier vector domain leads to a neat connection between clustering and the compressive sampling (CS) paradigm [5]. Building on this model, an outlier-aware clustering methodology is developed for clustering both from the deterministic (K-means), and the probabilistic (GMMs) perspectives.

The second contribution of this work comprises various iterative clustering algorithms developed for robust hard K-means, soft K-means, and GMM-based clustering (Section III). The algorithms are based on a block coordinate descent (BCD) iteration and yield closed-form updates for each set of optimization variables. In particular, estimating the outliers boils down to solving a group-Lasso problem [33], whose solution is computed in closed form. The novel robust clustering algorithms operate at an affordable computational complexity of the same order as the one for their non-robust counterparts.

Several contemporary applications in bioinformatics, (social) network analysis, image processing, and machine learning call for outlier-aware clustering of high-dimensional data, or involve nonlinearly separable clusters. To accommodate these clustering needs, the novel algorithms are kernelized in Section IV; and this is the third contribution of our work. The assumed model not only enables such a kernelization for both K-means and the probabilistic setups, but it also yields iterative algorithms with closed-form updates. In Section V, the algorithms developed are tested using synthetic as well as real datasets from handwritten digit recognition systems and social networks. The results corroborate the effectiveness of the methods. Conclusions are drawn in Section VI.

Notation: Lower-(upper-)case boldface letters are reserved for column vectors (matrices), and calligraphic letters for sets; $(\cdot)^T$ denotes transposition; $\mathbb{N}_N$ the set of naturals $\{1,\ldots,N\}$; $\mathbf{0}_p$ ($\mathbf{1}_p$) the $p\times 1$ vector of all zeros (ones); $\mathbf{I}_p$ the $p\times p$ identity matrix; $\mathrm{diag}(x_1,\ldots,x_p)$ a $p\times p$ diagonal matrix with diagonal entries $x_1,\ldots,x_p$; $\mathrm{range}(\mathbf{X})$ the range space of matrix $\mathbf{X}$; $\mathbb{E}[\cdot]$ denotes the expectation operator; $\mathcal{N}(\mathbf{x};\mathbf{m},\mathbf{\Sigma})$ denotes the multivariate Gaussian pdf with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$ evaluated at $\mathbf{x}$; $\|\mathbf{x}\|_{\mathbf{A}}:=\sqrt{\mathbf{x}^T\mathbf{A}\mathbf{x}}$ for a positive semidefinite matrix $\mathbf{A}$; $\|\mathbf{x}\|_p:=(\sum_{i=1}^n|x_i|^p)^{1/p}$ for $p\geq 1$ stands for the ℓ_p-norm in $\mathbb{R}^n$.

II. Sparsity-Aware Clustering: Context and Criteria

After reviewing the clustering task, a model pertinent to outlier-contaminated data is introduced next. Building on this model, robust approaches are developed for K-means (Section II-A) as well as probabilistic clustering (Section II-B).

A. K-means Clustering

Given a set of p-dimensional vectors $\mathcal{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, let $\{\mathcal{X}_1, \ldots, \mathcal{X}_C\}$ be a *partition* of $\mathcal{X}$ to C subsets (clusters) $\mathcal{X}_c \subset \mathcal{X}$ for $c \in \mathbb{N}_C$, which are collectively exhaustive, mutually exclusive, and non-empty. Partitional clustering seeks a partition of $\mathcal{X}$ such that two vectors assigned to the same cluster are closer to each other in some well-defined sense, such as the Euclidean distance, than to vectors assigned to other clusters.

Among partitional clustering methods, K-means is one of the most widely used with well-documented merits and a long history [4]. In the K-means setup, a centroid $\mathbf{m}_c \in \mathbb{R}^p$ is introduced per cluster $\mathcal{X}_c$. Then, instead of comparing distances between pairs of points in $\mathcal{X}$, the point-centroid distances $\|\mathbf{x}_n - \mathbf{m}_c\|_2$ are considered. Moreover, for each input vector $\mathbf{x}_n$, K-means introduces the unknown memberships u_{nc} for $c \in \mathbb{N}_C$, defined to be 1 when $\mathbf{x}_n \in \mathcal{X}_c$, and 0 otherwise. To guarantee a valid partition, the membership coefficients apart from being binary **(c1)**: $u_{nc} \in \{0, 1\}$; they should also satisfy the constraints **(c2)**: $\sum_{n=1}^{N} u_{nc} > 0$, for all c, to preclude empty clusters; and **(c3)**: $\sum_{c=1}^{C} u_{nc} = 1$, for all n, so that each vector is assigned to a cluster.

The K-means clustering task can be then posed as that of finding the centroids $\{\mathbf{m}_c\}_{c=1}^{C}$ and the cluster assignments u_{nc}'s by solving the optimization problem

$$\min_{\{\mathbf{m}_c\},\{u_{nc}\}} \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc} \|\mathbf{x}_n - \mathbf{m}_c\|_2^2 \text{ subject to (c1)-(c3).} \tag{1}$$

However, problem (1) is known to be NP-hard, even for $C = 2$ [7]. Practically, a suboptimal solution is pursued using the celebrated K-means algorithm. This algorithm drops the (c2) constraint, which is checked in a post-processing step instead. Then, it alternately minimizes the cost in (1) wrt one of the set of variables $\{\mathbf{m}_c\}$ and $\{u_{nc}\}$, while keeping the other one fixed, and iterates. K-means iterations are guaranteed to converge to a stationary point of (1) [28].

To better motivate and further understand the pros and cons of K-means clustering, it is instructive to postulate a pertinent data model. Such a model assumes that the input vectors can be expressed as $\mathbf{x}_n = \sum_{c=1}^{C} u_{nc}\mathbf{m}_c + \mathbf{v}_n$, where $\mathbf{v}_n$ is a zero-mean vector capturing the deviation of $\mathbf{x}_n$ from its associated centroid $\mathbf{m}_c$. It is easy to see that under (c1)-(c3), the minimizers of (1) offer merely a blind

least-squares (LS) fit of the data $\{\mathbf{x}_n\}_{n=1}^N$ respecting the cluster assignment constraints. However, such a simplistic, yet widely applicable model, does not take into account *outliers*; that is points $\mathbf{x}_n$ violating the assumed model. This fact paired with the sensitivity of the LS cost to large residuals explain K-means' vulnerability to outliers [8].

To robustify K-means, consider the following data model which explicitly accounts for outliers

$$\mathbf{x}_n = \sum_{c=1}^{C} u_{nc}\mathbf{m}_c + \mathbf{o}_n + \mathbf{v}_n, \quad n \in \mathbb{N}_N \tag{2}$$

where the outlier vector $\mathbf{o}_n$ is defined to be deterministically nonzero if $\mathbf{x}_n$ corresponds to an outlier, and $\mathbf{0}_p$ otherwise. The unknowns $\{u_{nc}, \mathbf{m}_c, \mathbf{o}_n\}$ in (2) can now be estimated using the LS approach as the minimizers of $\sum_{n=1}^N \left\|\mathbf{x}_n - \sum_{c=1}^C u_{nc}\mathbf{m}_c - \mathbf{o}_n\right\|_2^2$, which are the maximum likelihood (ML) estimates if $\mathbf{v}_n \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_p)$. Even if u_{nc}'s were known, estimating $\{\mathbf{m}_c\}$ and $\{\mathbf{o}_n\}$ based solely on $\{\mathbf{x}_n\}$ would be an under-determined problem. The key observation here is that most of the $\{\mathbf{o}_n\}$ are zero. This motivates the following criterion for clustering and identification of at most $s \in \mathbb{N}_N$ outliers

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}\in\mathcal{U}_1} \sum_{n=1}^{N} \left\|\mathbf{x}_n - \sum_{c=1}^{C} u_{nc}\mathbf{m}_c - \mathbf{o}_n\right\|_2^2 \text{ s. to } \sum_{n=1}^{N} \mathrm{I}\left(\|\mathbf{o}_n\|_2 > 0\right) \leq s \tag{3}$$

where $\mathbf{M} := [\mathbf{m}_1 \ \cdots \ \mathbf{m}_C]$, $\mathbf{O} := [\mathbf{o}_1 \ \cdots \ \mathbf{o}_N]$, $\mathbf{U} \in \mathbb{R}^{N\times C}$ denotes the membership matrix with entries $[\mathbf{U}]_{n,c} := u_{nc}$, $\mathcal{U}_1$ is the set of all $\mathbf{U}$ matrices satisfying (c1) and (c3), and $\mathrm{I}(\cdot)$ denotes the indicator function. Due to (c1) and (c3), each summand in the cost of (3) can be rewritten as $\sum_{c=1}^C u_{nc}\|\mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n\|_2^2$; and the Lagrangian form of (3) as

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}\in\mathcal{U}_1} \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}\left\|\mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n\right\|_2^2 + \lambda \sum_{n=1}^{N} \mathrm{I}\left(\|\mathbf{o}_n\|_2 > 0\right) \tag{4}$$

where $\lambda \geq 0$ is an outlier-controlling parameter. For $\lambda = 0$, $\mathbf{o}_n$ should be set equal to the generally nonzero value $\mathbf{x}_n - \mathbf{m}_c$ for any c and yield a zero optimum cost, in which case all $\mathbf{x}_n$'s are declared as outliers. When $\lambda \to \infty$, the optimum $\mathbf{o}_n$'s are zero, all the $\mathbf{x}_n$'s are deemed outlier free, and the problem in (4) reduces to the K-means cost in (1). This reduction of the NP-hard K-means problem to an instance of the problem in (4) establishes the NP-hardness of the latter.

Along the lines of K-means, similar iterations could be pursued for suboptimally solving (4). However, such iterations cannot provide any convergence guarantees due to the discontinuity of the indicator function at zero. Aiming at a practically feasible solver of (4), consider first that $\mathbf{U} \in \mathcal{U}_1$ is given. The optimization wrt $\{\mathbf{M}, \mathbf{O}\}$ remains non-convex due to $\sum_{n=1}^N \mathrm{I}(\|\mathbf{o}_n\|_2 > 0)$. Following the successful CS paradigm, where the ℓ_0-(pseudo)norm of a vector $\mathbf{x} \in \mathbb{R}^N$, defined as $\|\mathbf{x}\|_0 := \sum_{n=1}^N \mathrm{I}(|x_n| > 0)$, was

surrogated by its convex ℓ_1-norm $\|\mathbf{x}\|_1$, the problem in (4) is replaced by

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}\in\mathcal{U}_1} \quad \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}\,\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\|_2^2+\lambda\sum_{n=1}^{N}\|\mathbf{o}_n\|_2\,. \tag{5}$$

Our robust K-means approach is to minimize (5), which is convex in $\{\mathbf{M},\mathbf{O}\}$, but remains jointly non-convex. The algorithm for suboptimally solving the non-convex problem in (5) is postponed for Section III-A. Note that the minimization in (5) resembles the group Lasso criterion used for recovering a block-sparse vector in a linear regression setup [33]. This establishes an interesting link between robust clustering and CS. A couple of remarks are now in order.

Remark 1 (*Mahalanobis distance).* If the covariance matrix of $\mathbf{v}_n$ in (2) is known, say $\mathbf{\Sigma}$, the Euclidean distance in (3)-(5) can be replaced by the Mahalanobis distance $\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\|^2_{\mathbf{\Sigma}^{-1}}$.

Remark 2 (ℓ_1*-penalty for entry-wise outliers).* The regularization term $\sum_{n=1}^{N}\|\mathbf{o}_n\|_2$ in (5) enables identifying whole data vectors as outliers. Replacing it by $\sum_{n=1}^{N}\|\mathbf{o}_n\|_1$ enables recovery of outlying data entries instead of the whole vector. Iterative solvers for this case can be developed using the methodology presented in Section III; due to space limitations this case is not pursued here.

Constraints (c1) and (c3) in (1) entail *hard* membership assignments, meaning that each vector is assigned to a single cluster. However, *soft* clustering which allows each vector to partially belong to several clusters, can better identify overlapping clusters [1]. One way to obtain fractional memberships is via soft K-means. Soft K-means differs from hard K-means by (i) relaxing the binary-alphabet constraint (c1) to the box constraint **(c4):** $u_{nc}\in[0,1]$; and (ii) by raising the u_{nc}'s in (1) to the q-th power, where $q>1$ is a tuning parameter [1]. The robust soft K-means scheme proposed here amounts to replacing $\mathbf{x}_n$ with its outlier-compensated version $(\mathbf{x}_n-\mathbf{o}_n)$, and leveraging the sparsity of the $\{\mathbf{o}_n\}$'s. These steps lead to the following criterion

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}\in\mathcal{U}_2} \quad \sum_{n=1}^{N}\sum_{c=1}^{C} u_{nc}^{q}\left(\|\mathbf{x}_n-\mathbf{m}_c-\mathbf{o}_n\|_2^2+\lambda\,\|\mathbf{o}_n\|_2\right) \tag{6}$$

where $\mathcal{U}_2$ is the set of all $\mathbf{U}$ matrices satisfying (c3)-(c4). An algorithm for approximately solving (6) is presented in Section III-A. Note that a hard partition of $\mathcal{X}$ can still be obtained from the soft u_{nc} by assigning $\mathbf{x}_n$ to the $\hat{c}$-th cluster, where $\hat{c}:=\arg\max_c u_{nc}$.

B. Probabilistic Clustering

An alternative way to perform soft clustering is by following a probabilistic approach [31]. To this end, a mixture distribution model is postulated for $\mathbf{x}_n$, while $\{u_{nc}\}_{c=1}^{C}$ are now interpreted as unobserved

(latent) random variables. The centroids $\{\mathbf{m}_c\}_{c=1}^C$ are treated as deterministic parameters of the mixture distribution, and their ML estimates are subsequently obtained via the EM algorithm.

To account for outliers, probabilistic clustering is generalized to model (2). Suppose that the $\{\mathbf{x}_n\}$'s in (2) are i.i.d. drawn from a mixture model where the $\{\mathbf{o}_n\}$'s are deterministic parameters. The memberships $\mathbf{u}_n := [u_{n1} \cdots u_{nC}]^T$ are latent random vectors, corresponding to the rows of $\mathbf{U}$, and take values in $\{\mathbf{e}_1, \ldots, \mathbf{e}_C\}$, where $\mathbf{e}_c$ is the c-th column of $\mathbf{I}_C$. If $\mathbf{x}_n$ is drawn from the c-th mixture component, then $\mathbf{u}_n = \mathbf{e}_c$. Assume further that the class-conditional pdf's are Gaussian and modeled as $p(\mathbf{x}_n|\mathbf{u}_n = \mathbf{e}_c) = \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c+\mathbf{o}_n, \mathbf{\Sigma})$ for all n and c. This implies that $p(\mathbf{x}_n) = \sum_{c=1}^C \pi_c \, \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c+\mathbf{o}_n, \mathbf{\Sigma})$ with $\pi_c := \Pr(\mathbf{u}_n = \mathbf{e}_c)$. If the $\mathbf{x}_n$'s are independent, the log-likelihood of the input data is

$$L(\mathbf{X}; \boldsymbol{\pi}, \mathbf{M}, \mathbf{O}, \mathbf{\Sigma}) := \sum_{n=1}^N \log \left(\sum_{c=1}^C \pi_c \, \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c+\mathbf{o}_n, \mathbf{\Sigma}) \right) \tag{7}$$

where $\mathbf{X} := [\mathbf{x}_1 \ \cdots \ \mathbf{x}_N]$, and $\boldsymbol{\pi} := [\pi_1 \cdots \pi_C]^T$. Controlling the number of outliers (number of zero $\mathbf{o}_n$ vectors) suggests minimizing the *regularized* negative log-likelihood as

$$\min_{\mathbf{\Theta}} -L(\mathbf{X}; \mathbf{\Theta}) + \lambda \sum_{n=1}^N \|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}} \tag{8}$$

where $\mathbf{\Theta} := \{\boldsymbol{\pi} \in \mathcal{P}, \mathbf{M}, \mathbf{O}, \mathbf{\Sigma} \succ 0\}$ is the set of all model parameters, $\mathcal{P}$ is the probability simplex $\mathcal{P} := \{\boldsymbol{\pi} : \boldsymbol{\pi}^T\mathbf{1}{=}1 \text{ and } \boldsymbol{\pi} \geq \mathbf{0}\}$, and $\mathbf{\Sigma} \succ 0$ means that $\mathbf{\Sigma}$ is a positive definite matrix. An EM-based algorithm for solving (8) is derived in Section III-B. Having estimated the parameters of the likelihood, the posterior probabilities $\gamma_{nc} := \Pr(\mathbf{u}_n = \mathbf{e}_c|\mathbf{x}_n)$ can be readily obtained and interpreted as soft memberships.

Although modeling all class conditionals having a common covariance matrix $\mathbf{\Sigma}$ may seem restrictive, it guarantees that the GMM is well-posed, thereby avoiding spurious unbounded likelihood values [2, p. 433]. Specifically, it is easy to see that even if all $\mathbf{o}_n$'s are set to zero, the log-likelihood of a GMM with different covariance matrices $\mathbf{\Sigma}_c$ per mixture grows unbounded, e.g., by setting one of the $\mathbf{m}_c$'s equal to an $\mathbf{x}_n$ and letting $\mathbf{\Sigma}_c \to \mathbf{0}$ for that particular c. This possibility for unboundedness is also present in (8), and justifies the use of a common $\mathbf{\Sigma}$. But even with a common covariance matrix, the vectors $\mathbf{o}_n$ can drive the log-likelihood in (7) to infinity. Consider for example, any $(\mathbf{m}_c, \mathbf{o}_n)$ pair satisfying $\mathbf{x}_n = \mathbf{m}_c + \mathbf{o}_n$ and let $\mathbf{\Sigma} \to \mathbf{0}$. To make the problem of maximizing $L(\mathbf{X}; \mathbf{\Theta})$ well-posed, the $\|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}}$ regularizer is introduced. Note also that for $\lambda \to \infty$, the optimal $\mathbf{O}$ is zero and (8) reduces to the conventional MLE estimation of a GMM; whereas for $\lambda \to 0$, the cost in (8) becomes unbounded from below.

III. Robust Clustering Algorithms

Algorithms for solving the problems formulated in Section II are developed here. Section III-A focuses on the minimization of (6), while the minimization in (5) is obtained from (6) for $q = 1$. In Section III-B, an algorithm for minimizing (8) is derived based on the EM approach. Finally, modified versions of the new algorithms with enhanced resilience to outliers are pursued in Section III-C.

A. Robust (Soft) K-Means Algorithms

Consider first solving (6) for $q > 1$. Although the cost in (6) is jointly nonconvex, it is convex wrt each of $\mathbf{M}$, $\mathbf{O}$, and $\mathbf{U}$. To develop a suboptimum yet practical solver, the aforementioned per-variable convexity prompted us to devise a BCD algorithm, which minimizes the cost iteratively wrt each optimization variable while holding the other two variables fixed. Let $\mathbf{M}^{(t)}$, $\mathbf{O}^{(t)}$, and $\mathbf{U}^{(t)}$ denote the tentative solutions found at the t-th iteration. Also, initialize $\mathbf{U}^{(0)}$ randomly in $\mathcal{U}_2$, and $\mathbf{O}^{(0)}$ to zero.

In the first step of the t-th iteration, (6) is optimized wrt $\mathbf{M}$ for $\mathbf{U} = \mathbf{U}^{(t-1)}$ and $\mathbf{O} = \mathbf{O}^{(t-1)}$. The optimization decouples over the $\mathbf{m}_c$'s, and every $\mathbf{m}_c^{(t)}$ is the closed-form solution of an LS problem as

$$\mathbf{m}_c^{(t)} = \frac{\sum_{n=1}^{N}(u_{nc}^{(t-1)})^q \left(\mathbf{x}_n - \mathbf{o}_n^{(t-1)}\right)}{\sum_{n=1}^{N}(u_{nc}^{(t-1)})^q}. \tag{9}$$

In the second step, the task is to minimize (6) wrt $\mathbf{O}$ for $\mathbf{U} = \mathbf{U}^{(t-1)}$ and $\mathbf{M} = \mathbf{M}^{(t)}$. The optimization problem decouples per index n, so that each $\mathbf{o}_n$ can be found as the minimizer of

$$\phi^{(t)}(\mathbf{o}_n) := \sum_{c=1}^{C}(u_{nc}^{(t-1)})^q \left(\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n\|_2^2 + \lambda\|\mathbf{o}_n\|_2\right). \tag{10}$$

The cost $\phi^{(t)}(\mathbf{o}_n)$ is convex but non-differentiable. However, its minimizer can be expressed in closed form as shown in the ensuing proposition.

Proposition 1. *The optimization problem in* (10) *is uniquely minimized by*

$$\mathbf{o}_n^{(t)} = \mathbf{r}_n^{(t)} \left[1 - \frac{\lambda}{2\|\mathbf{r}_n^{(t)}\|_2}\right]_+ \tag{11}$$

where $[x]_+ := \max\{x, 0\}$*, and* $\mathbf{r}_n^{(t)}$ *is defined as*

$$\mathbf{r}_n^{(t)} := \frac{\sum_{c=1}^{C}(u_{nc}^{(t-1)})^q \left(\mathbf{x}_n - \mathbf{m}_c^{(t)}\right)}{\sum_{c=1}^{C}(u_{nc}^{(t-1)})^q}. \tag{12}$$

Proof: Since $\sum_{c=1}^{C}(u_{nc}^{(t)})^q > 0$ for all n and t due to (c3), the first summand of $\phi^{(t)}(\mathbf{o}_n)$ in (10) is a strictly convex function of $\mathbf{o}_n$. Hence, $\phi^{(t)}(\mathbf{o}_n)$ is a strictly convex function too and its minimizer is unique. Then, recall that a vector $\mathbf{o}_n^{(t)}$ is a minimizer of (10) if and only if $\mathbf{0} \in \partial\phi^{(t)}(\mathbf{o}_n^{(t)})$, where $\partial\phi^{(t)}(\mathbf{o}_n)$ is the sub-differential of $\phi^{(t)}(\mathbf{o}_n)$. For $\mathbf{o}_n \neq \mathbf{0}$, where the cost in (10) is differentiable, $\partial\phi^{(t)}(\mathbf{o}_n)$ is simply the gradient $-2\sum_{c=1}^{C}(u_{nc}^{(t-1)})^q\left(\mathbf{x}_n - \mathbf{m}_c - \left(1+\frac{\lambda}{2\|\mathbf{o}_n\|_2}\right)\mathbf{o}_n\right)$. At $\mathbf{o}_n = \mathbf{0}$, the sub-differential of the ℓ_2-norm $\|\mathbf{o}_n\|_2$ is the set of vectors $\{\mathbf{v}_n : \|\mathbf{v}_n\|_2 \leq 1\}$ by definition, and then the sub-differential of $\phi^{(t)}(\mathbf{o}_n)$ is $\partial\phi^{(t)}(\mathbf{o}_n) = \left\{-2\sum_{c=1}^{C}(u_{nc}^{(t-1)})^q\left(\mathbf{x}_n - \mathbf{m}_c - \frac{\lambda}{2}\mathbf{v}_n\right) : \ \|\mathbf{v}_n\|_2 \leq 1\right\}$.

When the minimizer $\mathbf{o}_n^{(t)}$ is nonzero, the condition $\mathbf{0} \in \partial\phi^{(t)}(\mathbf{o}_n^{(t)})$ implies

$$\left(1+\frac{\lambda}{2\|\mathbf{o}_n^{(t)}\|_2}\right)\mathbf{o}_n^{(t)} = \mathbf{r}_n^{(t)} \tag{13}$$

where $\mathbf{r}_n^{(t)}$ has been defined in (12). Equation (13) reveals that $\mathbf{o}_n^{(t)}$ is a positively scaled version of $\mathbf{r}_n^{(t)}$. The scaling can be readily found by taking the ℓ_2-norm on both sides of (13), i.e., $\|\mathbf{o}_n^{(t)}\|_2 = \|\mathbf{r}_n^{(t)}\|_2 - \lambda/2$, which is valid for $\|\mathbf{r}_n^{(t)}\|_2 > \lambda/2$. Substituting this back to (13), yields $\mathbf{o}_n^{(t)} = \mathbf{r}_n^{(t)}\left(1-\frac{\lambda}{2\|\mathbf{r}_n^{(t)}\|_2}\right)$.

For $\mathbf{o}_n^{(t)} = \mathbf{0}$, there exists a $\mathbf{v}_n^{(t)}$ for which $\|\mathbf{v}_n^{(t)}\|_2 \leq 1$ and $\mathbf{v}_n^{(t)} = (2/\lambda)\mathbf{r}_n^{(t)}$. This is possible when $\|\mathbf{r}_n^{(t)}\|_2 \leq \lambda/2$. These two cases for the minimizer of (10) are compactly expressed via (11). ■

The update for $\mathbf{o}_n^{(t)}$ in (11) reveals two interesting points: (i) the cost $\phi^{(t)}(\mathbf{o}_n)$ indeed favors zero minimizers; and (ii) the number of outliers is controlled by λ. After updating vector $\mathbf{r}_n^{(t)}$, its norm is compared against the threshold $\lambda/2$. If $\|\mathbf{r}_n^{(t)}\|_2$ exceeds $\lambda/2$, vector $\mathbf{x}_n$ is deemed an outlier, and it is compensated by a nonzero $\mathbf{o}_n^{(t)}$. Otherwise, $\mathbf{o}_n^{(t)}$ is set to zero and $\mathbf{x}_n$ is clustered as a regular point.

During the last step of the t-th iteration, (6) is minimized over $\mathbf{U} \in \mathcal{U}_2$ for $\mathbf{M} = \mathbf{M}^{(t)}$ and $\mathbf{O} = \mathbf{O}^{(t)}$. Similar to the conventional soft K-means, the minimizer is available in closed form as [1]

$$u_{nc}^{(t)} = \left[\sum_{c'=1}^{C}\left(\frac{\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n^{(t)}\|_2^2 + \lambda\|\mathbf{o}_n^{(t)}\|_2}{\|\mathbf{x}_n - \mathbf{m}_{c'}^{(t)} - \mathbf{o}_n^{(t)}\|_2^2 + \lambda\|\mathbf{o}_n^{(t)}\|_2}\right)^{\frac{1}{q-1}}\right]^{-1}. \tag{14}$$

Regarding the robust hard K-means, a similar BCD approach for solving (5) leads to updating $\mathbf{M}^{(t)}$ and $\mathbf{O}^{(t)}$ via (9), and (11)-(12) for $q = 1$. Updating $\mathbf{U}^{(t)}$ boils down to the minimum-distance rule

$$u_{nc}^{(t)} = \begin{cases} 1 & , \ c = \arg\min_{c'} \|\mathbf{x}_n - \mathbf{m}_{c'}^{(t)} - \mathbf{o}_n^{(t)}\|_2 \\ 0 & , \ \text{otherwise} \end{cases}. \tag{15}$$

Note that (15) is the limit case of (14) for $q \to 1^+$.

The robust K-means (RKM) algorithm is tabulated as Algorithm 1. RKM is terminated when $\|\mathbf{M}^{(t)} - \mathbf{M}^{(t-1)}\|_F / \|\mathbf{M}^{(t)}\|_F \leq \epsilon_s$, where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and ϵ_s is a small positive threshold, e.g., $\epsilon_s = 10^{-6}$. The computational resources needed by RKM are summarized next.

Algorithm 1 Robust K-means

Require: Input data matrix $\mathbf{X}$, number of clusters C, $q \geq 1$, and $\lambda > 0$.

1: Initialize $\mathbf{O}^{(0)}$ to zero and $\mathbf{U}^{(0)}$ randomly in $\mathcal{U}_2$.
2: **for** $t = 1, 2, \ldots$ **do**
3: Update $\mathbf{M}^{(t)}$ via (9).
4: Update $\mathbf{O}^{(t)}$ via (11)-(12).
5: Update $\mathbf{U}^{(t)}$ via (14) ($q > 1$) or (15) ($q = 1$).
6: **end for**

Remark 3 (*Computational complexity of RKM).* Suppose for concreteness that: **(as1)** the number of clusters is small, e.g., $C < p$; and **(as2)** the number of points is much larger than the input dimension, i.e., $N \gg p$. When (as2) does not hold, a modification of RKM is developed in Section IV. Under (as1)-(as2), the conventional K-means algorithm performs $\mathcal{O}(NCp)$ scalar operations per iteration, and requires storing $\mathcal{O}(Np)$ scalar variables. For RKM, careful counting shows that the per iteration time-complexity is maintained at $\mathcal{O}(NCp)$: (14) requires computing the NC Euclidean distances $\|\mathbf{x}_n - \mathbf{m}_c^{(t-1)} - \mathbf{o}_n^{(t-1)}\|_2^2$ and the N norms $\|\mathbf{o}_n^{(t-1)}\|_2$ which is $\mathcal{O}(NCp)$; $\mathbf{m}_c^{(t)}$'s are updated in $\mathcal{O}(NCp)$; while (11)-(12) entail $\mathcal{O}(NCp)$ operations. Further, the memory requirements of RKM are of the same order as those for K-means. Note also that the additional $N \times p$ matrix $\mathbf{O}$ can be stored using sparse structures.

The RKM iterations are convergent under mild conditions. This follows because the sequence of cost function values is non-increasing. Since the cost is bounded below, the function value sequences are guaranteed to converge. Convergence of the RKM iterates is characterized in the following proposition.

Proposition 2. *The RKM algorithm for $q \geq 1$ converges to a coordinate-wise minimum of* (6)*. Moreover, the hard RKM algorithm ($q = 1$) converges to a local minimum of* (5)*.*

Proof: By defining $f_s(c)$ as being zero when the Boolean argument c is true, and ∞ otherwise, the problem in (6) can be written in the unconstrained form

$$\min_{\mathbf{M},\mathbf{O},\mathbf{U}} \sum_{n=1}^{N} \sum_{c=1}^{C} u_{nc}^q \left(\|\mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n\|_2^2 + \lambda \|\mathbf{o}_n\|_2 \right) + f_s(\mathbf{U} \in \mathcal{U}_2). \tag{16}$$

The cost in (16), call it $f(\mathbf{M}, \mathbf{O}, \mathbf{U})$, is a proper and lower semi-continuous function, which implies that its non-empty level sets are closed. Also, since f is coercive, its level sets are bounded. Hence, the non-empty level sets of f are compact. For $q > 1$, function $f(\mathbf{M}, \mathbf{O}, \mathbf{U})$ has a unique minimizer per

optimization block variable $\mathbf{M}$, $\mathbf{O}$, and $\mathbf{U}$. Then, convergence of the RKM algorithm to a coordinate-wise minimum point of (6) follows from [29, Th. 4.1(c)].

When $q = 1$, define the first summand in (16) as $f_0(\mathbf{M}, \mathbf{O}, \mathbf{U}) := \sum_{n=1}^{N} \sum_{c=1}^{C} u_{nc} \|\mathbf{x}_n - \mathbf{m}_c - \mathbf{o}_n\|_2^2$, which is the differentiable part of f. Function f_0 has an open domain, and the remaining non-differentiable part of f is separable wrt the optimization blocks. Hence, again by [29, Th. 4.1(c)], the RKM algorithm with $q = 1$ converges to a local minimum $(\mathbf{M}^*, \mathbf{O}^*, \mathbf{U}^*)$ of (6).

It has been shown so far that for $q = 1$, a BCD iteration converges to a local minimum of (6). The BCD step for updating $\mathbf{U}$ is the hard rule in (15). Hence, this BCD algorithm (i) yields a $\mathbf{U}^*$ with binary entries, and (ii) essentially implements the BCD updates for solving (5). Since a local minimum of (6) with binary assignments is also a local minimum of (5), the claim of the proposition follows. ■

B. Robust Probabilistic Clustering Algorithm

An EM approach is developed in this subsection to carry out the minimization in (8). If $\mathbf{U}$ were known, the model parameters $\boldsymbol{\Theta}$ could be estimated by minimizing the regularized negative log-likelihood of the *complete data* $(\mathbf{X}, \mathbf{U})$; that is,

$$\min_{\boldsymbol{\Theta}} \; -L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta}) + \lambda \sum_{n=1}^{N} \|\mathbf{o}_n\|_{\boldsymbol{\Sigma}^{-1}} \tag{17}$$

where

$$L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta}) := \sum_{n=1}^{N} \sum_{c=1}^{C} u_{nc} \left(\log \pi_c + \log \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c + \mathbf{o}_n, \boldsymbol{\Sigma}) \right). \tag{18}$$

But since $\mathbf{U}$ is not observed, the cost in (17) is suboptimally minimized by iterating the two steps of the EM method. Let $\boldsymbol{\Theta}^{(t)}$ denote the model parameter values at the t-th iteration. During the E-step of the t-th iteration, the expectation $Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(t-1)}) := \mathbb{E}_{\mathbf{U}|\mathbf{X}; \boldsymbol{\Theta}^{(t-1)}} \left[L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta}) \right]$ is evaluated. Since $L(\mathbf{X}, \mathbf{U}; \boldsymbol{\Theta})$ is a linear function of $\mathbf{U}$, and u_{nc}'s are binary random variables, it follows that

$$Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(t-1)}) = \sum_{n=1}^{N} \sum_{c=1}^{C} \gamma_{nc}^{(t)} \left(\log \pi_c + \log \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c + \mathbf{o}_n, \boldsymbol{\Sigma}) \right) \tag{19}$$

where $\gamma_{nc}^{(t)} := \Pr\left(\mathbf{u}_n = \mathbf{e}_c | \mathbf{x}_n; \boldsymbol{\Theta}^{(t-1)} \right)$. Using Bayes' rule, the posterior probabilities $\gamma_{nc}^{(t)}$ are evaluated in closed form as

$$\gamma_{nc}^{(t)} = \frac{\pi_c^{(t-1)} \mathcal{N}(\mathbf{x}_n; \mathbf{m}_c^{(t-1)} + \mathbf{o}_n^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})}{\sum_{c'=1}^{C} \pi_{c'}^{(t-1)} \mathcal{N}(\mathbf{x}_n; \mathbf{m}_{c'}^{(t-1)} + \mathbf{o}_n^{(t-1)}, \boldsymbol{\Sigma}^{(t-1)})}. \tag{20}$$

During the M-step, $\boldsymbol{\Theta}^{(t)}$ is updated as

$$\boldsymbol{\Theta}^{(t)} = \arg\min_{\boldsymbol{\Theta}} \; -Q(\boldsymbol{\Theta}; \boldsymbol{\Theta}^{(t-1)}) + \lambda \sum_{n=1}^{N} \|\mathbf{o}_n\|_{\boldsymbol{\Sigma}^{-1}}. \tag{21}$$

A BCD strategy that updates each set of the parameters in $\mathbf{\Theta}$ one at a time with all other ones fixed, is described next. First, the cost in (21) is minimized wrt $\boldsymbol{\pi}$. Given that $\sum_{c=1}^{C}\gamma_{nc}^{(t)}=1$ for all n, it is easy to check that the minimizer of $-\sum_{n=1}^{N}\sum_{c=1}^{C}\gamma_{nc}^{(t)}\log\pi_c$ over $\mathcal{P}$ is found in closed form as

$$\pi_c^{(t)}=\frac{1}{N}\sum_{n=1}^{N}\gamma_{nc}^{(t)} \quad \text{for all} \quad c\in\mathbb{N}_C. \tag{22}$$

Subsequently, (21) is minimized wrt $\mathbf{M}$ while $\boldsymbol{\pi}$, $\mathbf{O}$, and $\mathbf{\Sigma}$ are set respectively to $\boldsymbol{\pi}^{(t)}$, $\mathbf{O}^{(t-1)}$, and $\mathbf{\Sigma}^{(t-1)}$. The centroids are updated as the minimizers of a weighted LS cost yielding

$$\mathbf{m}_c^{(t)}=\frac{\sum_{n=1}^{N}\gamma_{nc}^{(t)}\left(\mathbf{x}_n-\mathbf{o}_n^{(t-1)}\right)}{\sum_{n=1}^{N}\gamma_{nc}^{(t)}} \quad \text{for all} \quad c\in\mathbb{N}_C. \tag{23}$$

Then, (21) is minimized wrt $\mathbf{O}$ while keeping the rest of the model parameters fixed to their already updated values. This optimization decouples over n, and one has to solve

$$\min_{\mathbf{o}_n}\sum_{c=1}^{C}\frac{\gamma_{nc}^{(t)}}{2}\|\mathbf{x}_n-\mathbf{m}_c^{(t)}-\mathbf{o}_n\|^2_{(\mathbf{\Sigma}^{(t-1)})^{-1}}+\lambda\|\mathbf{o}_n\|_{(\mathbf{\Sigma}^{(t-1)})^{-1}} \tag{24}$$

for all $n\in\mathbb{N}_N$. For a full covariance $\mathbf{\Sigma}$, (24) can be solved as a second-order cone program. For the case of *spherical* clusters, i.e., $\mathbf{\Sigma}=\sigma^2\mathbf{I}_p$, solving (24) simplifies considerably. Specifically, the cost can then be written as $\sum_{c=1}^{C}\gamma_{nc}^{(t)}\|\mathbf{x}_n-\mathbf{m}_c^{(t)}-\mathbf{o}_n\|_2^2+2\lambda\sigma^{(t-1)}\|\mathbf{o}_n\|_2$, which is similar to the cost in (10) for $q=1$, and for an appropriately scaled λ. Building on the solution of (10), the $\mathbf{o}_n$'s are updated as

$$\mathbf{o}_n^{(t)}=\mathbf{r}_n^{(t)}\left[1-\frac{\lambda\sigma^{(t-1)}}{\|\mathbf{r}_n^{(t)}\|_2}\right]_+ \tag{25}$$

after redefining the residual vector as $\mathbf{r}_n^{(t)}:=\sum_{c=1}^{C}\gamma_{nc}^{(t)}(\mathbf{x}_n-\mathbf{m}_c^{(t)})$ in lieu of (12). Interestingly, the thresholding rule of (25) shows that $\sigma^{(t-1)}$ affects the detection of outliers. In fact, in this probabilistic setting, the threshold for outlier identification is proportional to the value of the outlier-compensated standard deviation estimate and, hence, it is adapted to the empirical distribution of the data.

The M-step is concluded by minimizing (21) wrt $\mathbf{\Sigma}$ for $\boldsymbol{\pi}=\boldsymbol{\pi}^{(t)}$, $\mathbf{M}=\mathbf{M}^{(t)}$, and $\mathbf{O}=\mathbf{O}^{(t)}$, i.e.,

$$\min_{\mathbf{\Sigma}\succ 0}\sum_{n=1}^{N}\sum_{c=1}^{C}\frac{\gamma_{nc}^{(t)}}{2}\|\mathbf{x}_n-\mathbf{m}_c^{(t)}-\mathbf{o}_n^{(t)}\|^2_{\mathbf{\Sigma}^{-1}}+\frac{N}{2}\log\det\mathbf{\Sigma}+\lambda\sum_{n=1}^{N}\|\mathbf{o}_n^{(t)}\|_{\mathbf{\Sigma}^{-1}}. \tag{26}$$

For a generic $\mathbf{\Sigma}$, (26) must be solved numerically, e.g., via gradient descent or interior point methods. Considering *spherical* clusters for simplicity, the first order optimality condition for (26) requires solving a quadratic equation in $\sigma^{(t)}$. Ignoring the negative root of this equation, $\sigma^{(t)}$ is found as

$$\sigma^{(t)}=\frac{\lambda}{2Np}\sum_{n=1}^{N}\|\mathbf{o}_n^{(t)}\|_2+\sqrt{\frac{1}{Np}\sum_{n=1}^{N}\sum_{c=1}^{C}\gamma_{nc}^{(t)}\|\mathbf{x}_n-\mathbf{m}_c^{(t)}-\mathbf{o}_n^{(t)}\|_2^2+\left(\frac{\lambda}{2Np}\sum_{n=1}^{N}\|\mathbf{o}_n^{(t)}\|_2\right)^2}. \tag{27}$$

Algorithm 2 Robust probabilistic clustering

Require: Input data matrix $\mathbf{X}$, number of clusters C, and parameter $\lambda > 0$.

1: Randomly initialize $\mathbf{M}^{(0)}$, $\boldsymbol{\pi}^{(0)} \in \mathcal{P}$, and set $\boldsymbol{\Sigma}^{(0)} = \delta \mathbf{I}_p$ ($\sigma^{(0)} = \sqrt{\delta}$) for $\delta > 0$, and $\mathbf{O}^{(0)}$ to zero.
2: **for** $t = 1, 2, \ldots$ **do**
3: Update $\gamma_{nc}^{(t)}$ via (20) for all n, c.
4: Update $\boldsymbol{\pi}^{(t)}$ via (22).
5: Update $\mathbf{M}^{(t)}$ via (23).
6: Update $\mathbf{O}^{(t)}$ by solving (24) ((25)).
7: Update $\boldsymbol{\Sigma}^{(t)}$ ($\sigma^{(t)}$) via (26) ((27)).
8: **end for**

The robust probabilistic clustering (RPC) scheme is tabulated as Algorithm 2. For spherical clusters, its complexity remains $\mathcal{O}(NCp)$ operations per iteration, even though the constants involved are larger than those in the RKM algorithm. Similar to RKM, the RPC iterations are convergent under mild conditions. Convergence of the RPC iterates is established in the next proposition.

Proposition 3. *The RPC iterations converge to a coordinate-wise minimum of the log-likelihood in* (7)*.*

Proof: Combining the two steps of the EM algorithm, namely (20) and (21), it is easy to verify that the algorithm is equivalent to a sequence of BCD iterations for optimizing

$$\min_{\boldsymbol{\Gamma}, \boldsymbol{\Theta}'} \; - \sum_{n=1}^{N} \sum_{c=1}^{C} \gamma_{nc} \log \left(\frac{\pi_c \mathcal{N}\left(\mathbf{x}_n; \mathbf{m}_c + \mathbf{o}_n, \boldsymbol{\Sigma}\right)}{\gamma_{nc}} \right) + \lambda \sum_{n=1}^{N} \|\mathbf{o}_n\|_{\boldsymbol{\Sigma}^{-1}} + f_s(\boldsymbol{\Gamma} \in \mathcal{U}_2) + f_s(\boldsymbol{\pi} \in \mathcal{P}) + f_s(\boldsymbol{\Sigma} \succ 0) \tag{28}$$

where $\boldsymbol{\Theta}' := \{\boldsymbol{\pi}, \mathbf{M}, \mathbf{O}, \boldsymbol{\Sigma}\}$, the $N \times C$ matrix $\boldsymbol{\Gamma}$ has entries $[\boldsymbol{\Gamma}]_{n,c} := \gamma_{nc} > 0$, and as in (16) that $f_s(c)$ is zero when condition c is true, and ∞ otherwise. That the $\{\gamma_{nc}\}$ are positive follows after using Bayes' rule to deduce that $\gamma_{nc} \propto \pi_c \mathcal{N}\left(\mathbf{x}_n; \mathbf{m}_c + \mathbf{o}_n, \boldsymbol{\Sigma}\right)$ and noticing that (i) $\mathcal{N}\left(\mathbf{x}_n; \mathbf{m}_c + \mathbf{o}_n, \boldsymbol{\Sigma}\right)$ is positive for all $\mathbf{x}_n$, and (ii) all π_c must be positive so that the cost in (28) remains finite.

The objective function of this minimization problem is proper, bounded below, and lower semi-continuous implying, that its non-empty level sets are closed. Since this function is also coercive, its level sets are bounded. Hence, its non-empty level sets are compact. Moreover, the objective function has a unique minimizer for the optimization blocks $\boldsymbol{\pi}$, $\mathbf{M}$, and $\mathbf{O}$. In particular, the $\mathbf{M}$ block minimizer is unique since $\sum_{n=1}^{N} \gamma_{nc} > 0$, for all $c \in \mathbb{N}_C$. Then, by [29, Th. 4.1 (c)], the RPC algorithm converges to a coordinate-wise minimum point of (7). ■

Proposition 3 guarantees that the RPC iterations converge. However, since each non-differentiable term $\|\mathbf{o}_n\|_{\mathbf{\Sigma}^{-1}}$ involves two different optimization variables $\mathbf{\Sigma}$ and $\mathbf{o}_n$, the BCD iteration can be trapped at a coordinate-wise local minimum, which is not necessarily a local minimum of (8). Once the iterations have converged, the final γ_{nc}'s can be interpreted as soft cluster assignments, whereby hard assignments can be obtained via the maximum a posteriori detection rule, i.e., $\mathbf{x}_n \in \mathcal{X}_c$ for $c = \arg\max_{c'} \gamma_{nc'}$.

Remark 4 (*Selecting* λ)*.* Tuning λ is possible if additional information, e.g., on the percentage of outliers, is available. The robust clustering algorithm is ran for a decreasing sequence of λ values $\{\lambda_g\}$, using "warm starts" [11], until the expected number of outliers is identified. When solving for λ_g, warm start refers to the optimization variables initialized to the solution obtained for λ_{g-1}. Hence, running the algorithm over $\{\lambda_g\}$ becomes very efficient, because few BCD iterations per λ_g suffice for convergence.

C. Weighted Robust Clustering Algorithms

As already mentioned, the robust clustering methods presented so far approximate the discontinuous penalty $\mathrm{I}(\|\mathbf{o}_n\|_2 > 0)$ by $\|\mathbf{o}_n\|_2$, mimicking the CS paradigm in which $\mathrm{I}(|x| > 0)$ is surrogated by the convex function $|x|$. However, it has been argued that non-convex functions such as $\log(|x| + \epsilon)$ for a small $\epsilon > 0$ can offer tighter approximants of $\mathrm{I}(|x| > 0)$ [30]. This rationale prompted us to replace $\|\mathbf{o}_n\|_2$ in (5), (6), and (8), by the penalty $\log(\|\mathbf{o}_n\|_2 + \epsilon)$ to further enhance block sparsity in $\mathbf{o}_n$'s, and thereby improve resilience to outliers.

Altering the regularization term modifies the BCD algorithms only when minimizing wrt $\mathbf{O}$. This particular step remains decoupled across $\mathbf{o}_n$'s, but instead of the $\phi^{(t)}(\mathbf{o}_n)$ defined in (10), one minimizes

$$\phi_w^{(t)}(\mathbf{o}_n) := \sum_{c=1}^{C} (u_{nc}^{(t-1)})^q \left(\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n\|_2^2 + \lambda \log\left(\|\mathbf{o}_n\|_2 + \epsilon\right) \right) \tag{29}$$

that is no longer convex. The optimization in (29) is performed using a single iteration of the majorization-minimization (MM) approach[1] [21]. The cost $\phi_w^{(t)}(\mathbf{o}_n)$ is majorized by a function $f^{(t)}(\mathbf{o}_n; \mathbf{o}_n^{(t-1)})$, which means that $\phi_w^{(t)}(\mathbf{o}_n) \leq f^{(t)}(\mathbf{o}_n; \mathbf{o}_n^{(t-1)})$ for every $\mathbf{o}_n$ and $\phi_w^{(t)}(\mathbf{o}_n) = f^{(t)}(\mathbf{o}_n; \mathbf{o}_n^{(t-1)})$ when $\mathbf{o}_n = \mathbf{o}_n^{(t-1)}$. Then $f^{(t)}(\mathbf{o}_n; \mathbf{o}_n^{(t-1)})$ is minimized wrt $\mathbf{o}_n$ to obtain $\mathbf{o}_n^{(t)}$.

To find a majorizer for $\phi_w^{(t)}(\mathbf{o}_n)$, the concavity of the logarithm is exploited, i.e., the fact that $\log x \leq \log x_o + x/x_o - 1$ for any positive x and x_o. Applying the last inequality for the penalty and ignoring

[1]Note that the MM approach for minimizing $\phi_w^{(t)}(\mathbf{o}_n)$ at the t-th BCD iteration involves several internal MM iterations. Due to the external BCD iterations and to speed up the algorithm, a single MM iteration is performed per BCD iteration t.

the constant terms involved, we end up minimizing

$$f^{(t)}(\mathbf{o}_n;\mathbf{o}_n^{(t-1)}) := \sum_{c=1}^{C}(u_{nc}^{(t-1)})^q \left(\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n\|_2^2 + \lambda_n^{(t)}\|\mathbf{o}_n\|_2\right) \tag{30}$$

where $\lambda_n^{(t)} := \lambda/(\|\mathbf{o}_n^{(t-1)}\|_2 + \epsilon)$. Comparing (30) to (10) shows that the new regularization results in a weighted version of the original one. The only difference between the robust algorithms presented earlier and their henceforth termed *weighted* counterparts is the definition of λ. At iteration t, larger values for $\|\mathbf{o}_n^{(t-1)}\|_2$ lead to smaller thresholds in the thresholding rules (cf. (11), (25)), thereby making $\mathbf{o}_n$ more likely to be selected as nonzero. The weighted robust clustering algorithms initialize $\mathbf{o}_n^{(0)}$ to the associated $\mathbf{o}_n$ value the non-weighted algorithm converged to. Thus, to run the weighted RKM for a specific value of λ, the RKM needs to be run first. Then, weighted RKM is run with all the variables initialized to the values attained by RKM, but with the $\lambda_n^{(1)}$ as defined earlier.

The MM step combined with the BCD algorithms developed hitherto are convergent under mild assumptions. To see this, note that the sequences of objective values for the RKM and RPC algorithms are both non-increasing. Since the respective cost functions are bounded below, those sequences are guaranteed to converge. Characterizing the points and speed of convergence goes beyond the scope of this paper.

IV. Clustering High-Dimensional and Nonlinearly Separable Data

The robust clustering algorithms of Section III are kernelized here. The advantage of kernelization is twofold: (i) yields computationally efficient algorithms when dealing with high-dimensional data, and (ii) robustly identifies nonlinearly separable clusters.

A. Robust K-means for High-Dimensional Data

The robust clustering algorithms derived so far involve generally $\mathcal{O}(NCp)$ operations per iteration. However, several applications entail clustering relatively few but *high-dimensional* data in the presence of outliers. In imaging applications, one may wish to cluster $N = 500$ images of say $p = 800 \times 600 = 480,000$ pixels; while in DNA microarray data analysis, some tens of (potentially erroneous or rarely occurring) DNA samples are to be clustered based on their expression levels over thousands of genes [14]. In such clustering scenarios where $p \gg N$, an efficient method should avoid storing and processing p-dimensional vectors [27]. To this end, the algorithms of Section III are *kernelized* here [25]. It will be shown that these kernelized algorithms require $\mathcal{O}(N^3C)$ operations per iteration and $\mathcal{O}(N^2)$ space; hence, they are preferable when $p > N^2$. This kernelization not only offers processing savings in the

high-dimensional data regime, but also serves as the building module for identifying nonlinearly separable data clusters as pursued in the next subsection.

We focus on kernelizing the robust soft K-means algorithm; the kernelized robust hard K-means can then be derived after simple modifications. Consider the $N\times C$ matrix $\mathbf{U}_q$ with entries $[\mathbf{U}_q]_{nc}:=u_{nc}^q$, and the Grammian $\mathbf{K}:=\mathbf{X}^T\mathbf{X}$ formed by all pairwise inner products between the input vectors. Even though the cost for computing $\mathbf{K}$ is $\mathcal{O}(N^2p)$, it is computed only once. Note that the updates (9), (11), and (14) involve inner products between the p-dimensional vectors $\{\mathbf{o}_n,\mathbf{r}_n\}_{n=1}^N$, and $\{\mathbf{m}_c\}_{c=1}^C$. If $\{\mathbf{v}_i\in\mathbb{R}^p\}_{i=1}^2$ is a pair of any of these vectors, the cost for computing $\mathbf{v}_1^T\mathbf{v}_2$ is clearly $\mathcal{O}(p)$. But if at every BCD iteration the aforementioned vectors lie in $\mathrm{range}(\mathbf{X})$, i.e., if there exist $\{\mathbf{w}_i\in\mathbb{R}^N\}_{i=1}^2$ such that $\{\mathbf{v}_i=\mathbf{X}\mathbf{w}_i\}_{i=1}^2$, then $\mathbf{v}_1^T\mathbf{v}_2=\mathbf{w}_1^T\mathbf{K}\mathbf{w}_2$, and the inner product can be alternatively calculated in $\mathcal{O}(N^2)$.

Hinging on this observation, it is first shown that all the $p\times 1$ vectors involved indeed lie in $\mathrm{range}(\mathbf{X})$. The proof is by induction: if at the $(t-1)$-st iteration every $\mathbf{o}_n^{(t-1)}\in\mathrm{range}(\mathbf{X})$ and $\mathbf{U}^{(t-1)}\in\mathcal{U}_2$, it will be shown that $\mathbf{o}_n^{(t)}$, $\mathbf{m}_c^{(t)}$, $\mathbf{r}_n^{(t)}$ updated by RKM lie in $\mathrm{range}(\mathbf{X})$ as well.

Suppose that at the t-th iteration, the matrix $\mathbf{U}^{(t-1)}$ defining $\mathbf{U}_q^{(t-1)}$ is in $\mathcal{U}_2$, while there exists matrix $\mathbf{A}^{(t-1)}$ such that $\mathbf{O}^{(t-1)}=\mathbf{X}\mathbf{A}^{(t-1)}$. Then, the update of the centroids in (9) can be expressed as

$$\mathbf{M}^{(t)}=(\mathbf{X}-\mathbf{O}^{(t-1)})\mathbf{U}_q^{(t-1)}\,\mathrm{diag}^{-1}((\mathbf{U}_q^{(t-1)})^T\mathbf{1}_N)=\mathbf{X}\mathbf{B}^{(t)} \tag{31}$$

where

$$\mathbf{B}^{(t)}:=(\mathbf{I}_N-\mathbf{A}^{(t-1)})\mathbf{U}_q^{(t-1)}\,\mathrm{diag}^{-1}((\mathbf{U}_q^{(t-1)})^T\mathbf{1}_N). \tag{32}$$

Before updating $\mathbf{O}^{(t)}$, the residual vectors $\{\mathbf{r}_n\}$ must be updated via (12). Concatenating the residuals in $\mathbf{R}^{(t)}:=[\mathbf{r}_1^{(t)}\ldots\mathbf{r}_N^{(t)}]$, the update in (12) can be rewritten in matrix form as

$$\mathbf{R}^{(t)}=\mathbf{X}-\mathbf{M}^{(t)}(\mathbf{U}_q^{(t-1)})^T\,\mathrm{diag}^{-1}(\mathbf{U}_q^{(t-1)}\mathbf{1}_C)=\mathbf{X}\boldsymbol{\Delta}^{(t)} \tag{33}$$

where

$$\boldsymbol{\Delta}^{(t)}:=\mathbf{I}_N-\mathbf{B}^{(t)}(\mathbf{U}_q^{(t-1)})^T\,\mathrm{diag}^{-1}(\mathbf{U}_q^{(t-1)}\mathbf{1}_C). \tag{34}$$

From (11), every $\mathbf{o}_n^{(t)}$ is a scaled version of $\mathbf{r}_n^{(t)}$ and the scaling depends on $\|\mathbf{r}_n^{(t)}\|_2$. Based on (33), the latter can be readily computed as $\|\mathbf{r}_n^{(t)}\|_2=\sqrt{(\boldsymbol{\delta}_n^{(t)})^T\mathbf{K}\boldsymbol{\delta}_n^{(t)}}=\|\boldsymbol{\delta}_n^{(t)}\|_{\mathbf{K}}$, where $\boldsymbol{\delta}_n^{(t)}$ stands for the n-th column of $\boldsymbol{\Delta}^{(t)}$. Upon applying the thresholding operator, one arrives at the update

$$\mathbf{O}^{(t)}=\mathbf{X}\mathbf{A}^{(t)} \tag{35}$$

where the n-th column of $\mathbf{A}^{(t)}$ is given by

$$\boldsymbol{\alpha}_n^{(t)}=\boldsymbol{\delta}_n^{(t)}\left[1-\frac{\lambda}{2\|\boldsymbol{\delta}_n^{(t)}\|_{\mathbf{K}}}\right]_+,\quad\forall n. \tag{36}$$

Having proved the inductive step by (35), the argument is complete if and only if the outlier variables $\mathbf{O}$ are initialized as $\mathbf{O}^{(0)} = \mathbf{X}\mathbf{A}^{(0)}$ for some $\mathbf{A}^{(0)}$, including the practically interesting and meaningful initialization at zero. The result just proved can be summarized as follows.

Proposition 4. *By choosing $\mathbf{O}^{(0)} = \mathbf{X}\mathbf{A}^{(0)}$ for any $\mathbf{A}^{(0)} \in \mathbb{R}^{N\times N}$ and $\mathbf{U}^{(0)} \in \mathcal{U}_2$, the columns of the matrix variables $\mathbf{O}$, $\mathbf{M}$, and $\mathbf{R}$ updated by RKM all lie in $\mathrm{range}(\mathbf{X})$; i.e., there exist known $\mathbf{A}^{(t)}$, $\mathbf{B}^{(t)}$, and $\mathbf{\Delta}^{(t)}$, such that $\mathbf{O}^{(t)} = \mathbf{X}\mathbf{A}^{(t)}$, $\mathbf{M}^{(t)} = \mathbf{X}\mathbf{B}^{(t)}$, and $\mathbf{R}^{(t)} = \mathbf{X}\mathbf{\Delta}^{(t)}$ for all t.*

What remains to be kernelized are the updates for the cluster assignments. For the update step (14) or (15), we need to compute $\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n^{(t)}\|_2^2$ and $\|\mathbf{o}_n^{(t)}\|_2$. Given that $\mathbf{x}_n = \mathbf{X}\mathbf{e}_n$, where $\mathbf{e}_n$ denotes the n-th column of $\mathbf{I}_N$, and based on the kernelized updates (31) and (35), it is easy to verify that

$$\|\mathbf{x}_n - \mathbf{m}_c^{(t)} - \mathbf{o}_n^{(t)}\|_2^2 = \|\mathbf{X}(\mathbf{e}_n - \boldsymbol{\beta}_c^{(t)} - \boldsymbol{\alpha}_n^{(t)})\|_2^2 = \|\mathbf{e}_n - \boldsymbol{\beta}_c^{(t)} - \boldsymbol{\alpha}_n^{(t)}\|_{\mathbf{K}}^2 \tag{37}$$

for every n and c, where $\boldsymbol{\beta}_c^{(t)}$ is the c-th column of $\mathbf{B}^{(t)}$. As in (35), it follows that

$$\|\mathbf{o}_n^{(t)}\|_2 = \|\mathbf{X}\boldsymbol{\alpha}_n^{(t)}\|_2 = \|\boldsymbol{\alpha}_n^{(t)}\|_{\mathbf{K}}. \tag{38}$$

The kernelized robust K-means (KRKM) algorithm is summarized as Algorithm 3. As for RKM, the KRKM algorithm is terminated when $\|\mathbf{M}^{(t)} - \mathbf{M}^{(t-1)}\|_F / \|\mathbf{M}^{(t)}\|_F \leq \epsilon_s$ for a small $\epsilon_s > 0$. Based on (31) and exploiting standard linear algebra properties, the stopping condition can be equivalently expressed as $(\sum_{c=1}^{C} \|\boldsymbol{\beta}_c^{(t)} - \boldsymbol{\beta}_c^{(t-1)}\|_{\mathbf{K}}^2)/(\sum_{c=1}^{C} \|\boldsymbol{\beta}_c^{(t)}\|_{\mathbf{K}}^2) \leq \epsilon_s^2$.

Notice that this kernelized algorithm does not explicitly update $\mathbf{M}$, $\mathbf{R}$, or $\mathbf{O}$; actually, these variables are never processed. Instead, it updates $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{\Delta}$; while the clustering assignments are updated via (14), (37), and (38). Ignoring the cost for finding $\mathbf{K}$, the computations required by this algorithm are $\mathcal{O}(N^3C)$ per iteration, whereas the stored variables $\mathbf{A}$, $\mathbf{B}$, $\mathbf{\Delta}$, $\mathbf{U}_q$, and $\mathbf{K}$ occupy $\mathcal{O}(N^2)$ space. Note that if the centroids $\mathbf{M}$ are explicitly needed (e.g., for interpretative purposes or for clustering new input data), they can be acquired via (31) after KRKM has terminated.

B. Kernelized RKM for Nonlinearly Separable Clusters

One of the limitations of conventional K-means is that clusters should be of spherical or, more generally, ellipsoidal shape. By postulating the squared Euclidean distance as the similarity metric between vectors, the underlying clusters are tacitly assumed to be linearly separable. GMM-based clustering shares the same limitation. Kernel K-means has been proposed to overcome this obstacle [26] by mapping vectors $\mathbf{x}_n$ to a higher dimensional space $\mathcal{H}$ through the nonlinear function $\varphi : \mathbb{R}^p \to \mathcal{H}$. The mapped data

Algorithm 3 Kernelized RKM

Require: Grammian matrix $\mathbf{K} \succ 0$, number of clusters C, $q \geq 1$, and $\lambda > 0$.

1: Initialize $\mathbf{U}^{(0)}$ randomly in $\mathcal{U}_2$, and $\mathbf{A}^{(0)}$ to zero.
2: **for** $t = 1, 2, \ldots$ **do**
3: Update $\mathbf{B}^{(t)}$ from (32).
4: Update $\mathbf{\Delta}^{(t)}$ from (34).
5: Update $\mathbf{A}^{(t)}$ from (36).
6: Update $\mathbf{U}^{(t)}$ and $\mathbf{U}_q^{(t)}$ from (14) or (15), (37), and (38).
7: **end for**

$\{\varphi(\mathbf{x}_n)\}_{n=1}^N$ lie in the so-termed feature space which is of dimension $P > p$ or even infinite. The conventional K-means algorithm is subsequently applied in its kernelized version on the transformed data. Thus, linearly separable partitions in feature space yield nonlinearly separable partitions in the original space.

For an algorithm to be kernelizable, that is to be able to operate with inputs in feature space, the inner product between any two mapped vectors, i.e., $\varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m)$, should be known. For the non-kernelized versions of K-means, RKM, and RPC algorithms, where the linear mapping $\varphi(\mathbf{x}_n) = \mathbf{x}_n$ can be trivially assumed, these inner products are directly computable and stored at the (n, m)-th entry of the Grammian matrix $\mathbf{K} = \mathbf{X}^T\mathbf{X}$. When a nonlinear mapping is used, the so-termed kernel matrix $\mathbf{K}$ with entries $[\mathbf{K}]_{n,m} := \varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m)$ replaces the Grammian matrix and must be known. By definition, $\mathbf{K}$ must be positive semidefinite and can be employed for (robust) clustering, even when $\varphi(\mathbf{x}_n)$ is high-dimensional (cf. Section IV-A), infinite-dimensional, or even unknown [10].

Of particular interest is the case where $\mathcal{H}$ is a reproducing kernel Hilbert space. Then, the inner product in $\mathcal{H}$ is provided by a known kernel function $\kappa(\mathbf{x}_n, \mathbf{x}_m) := \varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m)$ [25, Ch. 3]. Typical kernels for vectorial input data are the polynomial, Gaussian, and the sigmoid ones; however, kernels can be defined for non-vectorial objects as well, such as strings or graphs [25].

After having RKM tailored to the high-dimensional input data regime in Section IV-A, handling arbitrary kernel functions is now straightforward. Knowing the input data $\mathbf{X}$ and the kernel function $\kappa(\mathbf{x}_n, \mathbf{x}_m)$, the kernel matrix can be readily computed as $[\mathbf{K}]_{n,m} = \kappa(\mathbf{x}_n, \mathbf{x}_m)$ for $n,\ m \in \mathbb{N}_N$. By using the kernel in lieu of the Grammian matrix, Algorithm 3 carries over readily to the nonlinear clustering regime.

As shown earlier, when clustering high-dimensional data, the centroids can be computed after the robust clustering algorithm has terminated via (31). This is not generally the case withrobust nonlinear clustering. For infinite-dimensional or even finite dimensional feature spaces, such as the one induced by a polynomial kernel, even if one is able to recover the centroid $\mathbf{m}_c \in \mathcal{H}$, its pre-image in the input space may not exist [25, Ch. 18].

C. Kernelized Robust Probabilistic Clustering

Similar to RKM, the RPC algorithm can be kernelized to (i) facilitate computationally efficient clustering of high-dimensional input data, and (ii) enable nonlinearly separable clustering. To simplify the presentation, the focus here is on the case of spherical clusters.

Kernelizing RPC hinders a major difference over the kernelization of RKM: the GMM and the RPC updates in Section III-B remain valid for $\{\varphi(\mathbf{x}_n) \in \mathcal{H}\}$ only when the feature space dimension P remains finite and known. The implication is elucidated as follows. First, updating the variance in (27) entails the underlying vector dimension p – which becomes P when it comes to kernelization. Second, the (outlier-aware) mixtures of Gaussians degenerate when it comes to modeling infinite-dimensional random vectors. To overcome this limitation, the notion of the empirical kernel map will be exploited [25, Ch. 2.2.6]. Given the fixed set of vectors in $\mathcal{X}$, instead of φ, it is possible to consider the empirical kernel map $\hat{\varphi} : \mathbb{R}^p \to \mathbb{R}^N$ defined as $\hat{\varphi}(\mathbf{x}) := (\mathbf{K}^{1/2})^{\dagger}[\kappa(\mathbf{x}_1, \mathbf{x}) \cdots \kappa(\mathbf{x}_N, \mathbf{x})]^T$, where $\mathbf{K}$ is the kernel matrix of the input data $\mathcal{X}$, and $(\cdot)^{\dagger}$ the Moore-Penrose pseudoinverse. The feature space $\hat{\mathcal{H}}$ induced by $\hat{\varphi}$ has finite dimensionality N. It can be also verified that $\hat{\varphi}^T(\mathbf{x}_n)\hat{\varphi}(\mathbf{x}_m) = \varphi^T(\mathbf{x}_n)\varphi(\mathbf{x}_m) = \kappa(\mathbf{x}_n, \mathbf{x}_m)$ for all $\mathbf{x}_n,\ \mathbf{x}_m \in \mathcal{X}$; hence, inner products in $\hat{\mathcal{H}}$ are readily computable through κ.

In the kernelized probabilistic setup, vectors $\{\hat{\varphi}(\mathbf{x}_n)\}_{n=1}^N$ are assumed drawn from a mixture of C multivariate Gaussian distributions with common covariance $\mathbf{\Sigma} = \sigma^2\mathbf{I}_N$ for all clusters. The EM-based updates of RPC in Section III-B remain valid after replacing the dimension p in (27) by N, and the input vectors $\{\mathbf{x}_n\}$ by $\{\hat{\varphi}(\mathbf{x}_n)\}$ with the critical property that one only needs to know the inner products $\hat{\varphi}^T(\mathbf{x}_n)\hat{\varphi}(\mathbf{x}_m)$ stored as the (n, m)-th entries of the kernel matrix $\mathbf{K}$. The kernelization procedure is similar to the one followed for RKM: first, the auxiliary matrices $\mathbf{A}^{(t)}$, $\mathbf{B}^{(t)}$, and $\mathbf{\Delta}^{(t)}$ are introduced. By randomly initializing $\sigma^{(0)}$, $\boldsymbol{\pi}^{(0)} \in \mathcal{P}$, $\mathbf{B}^{(0)} \in \mathbb{R}^{N\times C}$, and setting $\mathbf{A}^{(0)}$ to zero, it can be shown as in Proposition 4, that the kernelized RPC updates for $\mathbf{O}^{(t)}$, $\mathbf{M}^{(t)}$, and $\mathbf{R}^{(t)}$ have their columns lying in $\mathrm{range}(\mathbf{\Phi})$, where $\mathbf{\Phi} := [\hat{\varphi}(\mathbf{x}_1) \cdots \hat{\varphi}(\mathbf{x}_N)]$. Instead of the assignment matrix $\mathbf{U}$ in KRKM, the $N \times C$ matrix of posterior probability estimates $\mathbf{\Gamma}^{(t)}$ is used, where $[\mathbf{\Gamma}^{(t)}]_{n,c} := \gamma_{nc}^{(t)}$ satisfying $\mathbf{\Gamma}^{(t)}\mathbf{1}_C = \mathbf{1}_N\ \forall t$.

Algorithm 4 Kernelized RPC

Require: Grammian or kernel matrix $\mathbf{K} \succ 0$, number of clusters C, and $\lambda > 0$.

1: Randomly initialize $\sigma^{(0)}$, $\boldsymbol{\pi}^{(0)} \in \mathcal{P}$, and $\mathbf{B}^{(0)}$; and set $\mathbf{A}^{(0)}$ to zero.

2: **for** $t = 1, 2, \ldots$ **do**

3: Update $\boldsymbol{\Gamma}^{(t)}$ via (20) exploiting $\|\mathbf{x}_n - \mathbf{m}_c^{(t-1)} - \mathbf{o}_n^{(t-1)}\|_2^2 = \|\mathbf{e}_n - \boldsymbol{\beta}_c^{(t-1)} - \boldsymbol{\alpha}_n^{(t-1)}\|_{\mathbf{K}}^2$ for all n, c.

4: Update $\boldsymbol{\pi}^{(t)}$ as $\boldsymbol{\pi}^{(t)} = (\boldsymbol{\Gamma}^{(t)})^T \mathbf{1}_N / N$.

5: Update $\mathbf{B}^{(t)}$ as $\mathbf{B}^{(t)} = (\mathbf{I}_N - \mathbf{A}^{(t-1)})\boldsymbol{\Gamma}^{(t)} \operatorname{diag}^{-1}(N\boldsymbol{\pi}^{(t)})$.

6: Update $\boldsymbol{\Delta}^{(t)}$ as $\boldsymbol{\Delta}^{(t)} = \mathbf{I}_N - \mathbf{B}^{(t)}(\boldsymbol{\Gamma}^{(t)})^T$.

7: Update the columns of $\mathbf{A}^{(t)}$ as $\boldsymbol{\alpha}_n^{(t)} = \boldsymbol{\delta}_n^{(t)} \left[1 - \frac{\lambda \sigma^{(t-1)}}{\|\boldsymbol{\delta}_n^{(t)}\|_{\mathbf{K}}}\right]_+$ for all n.

8: Update $\sigma^{(t)}$ via (27) where p is replaced by N, using the ℓ_2-norms computed in Step 3, and exploiting $\|\mathbf{o}_n^{(t)}\|_2 = \|\boldsymbol{\alpha}_n^{(t)}\|_{\mathbf{K}}$ for all n.

9: **end for**

The kernelized RPC (KRPC) algorithm is summarized as Alg. 4. As with KRKM, its computations are $\mathcal{O}(N^3 C)$ per iteration, whereas the stored variables $\mathbf{A}$, $\mathbf{B}$, $\boldsymbol{\Delta}$, $\boldsymbol{\Gamma}$, $\boldsymbol{\pi}$, $\mathbf{K}$, and σ occupy $\mathcal{O}(N^2)$ space.

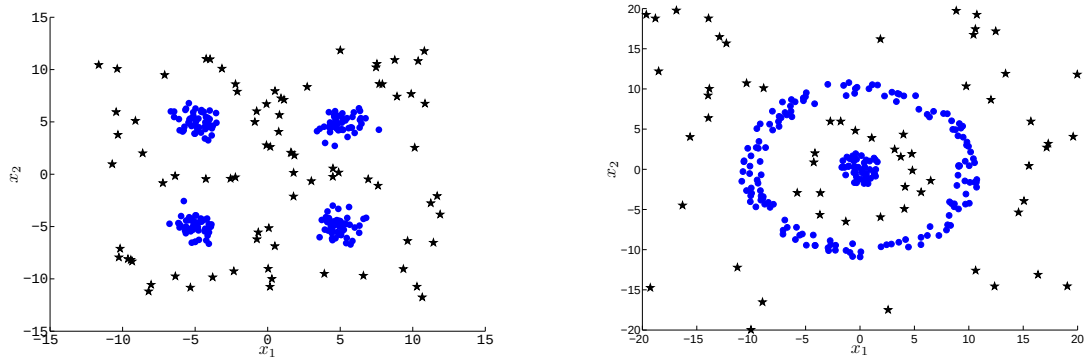
Remark 5 (*Reweighted kernelized algorithms).* Reweighted kernelized algorithms similar to the ones in Section III-C can be derived for KRKM and KRPC by a simple modification. In both cases, it suffices to introduce an iteration-dependent parameter $\lambda_n^{(t)} = \lambda / (\|\mathbf{o}_n^{(t-1)}\|_2 + \epsilon)$ per index n. Note that $\|\mathbf{o}_n\|_2$'s can be readily computed in terms of kernels as shown earlier.

V. Numerical Tests

Numerical tests illustrating the performance of the novel robust clustering algorithms on both synthetic and real datasets are shown in this section. Performance is assessed through their ability to identify outliers and the quality of clustering itself. The latter is measured using the adjusted rand index (ARI) between the partitioning found and the true partitioning of the data whenever the latter is available [17]. In each experiment, the parameter λ is tuned using the grid search outlined in Remark 4 over at most $1,000$ values. Thanks to the warm-start technique, the solution path for all grid points was computed in an amount of time comparable to the one used for solving for a specific value of λ.

A. Synthetic Datasets

Two synthetic datasets are used. The first one, shown in Fig. 1(a), consists of a random draw of 200 vectors from $C = 4$ bivariate Gaussian distributions (50 vectors per distribution), and 80 outlying vectors



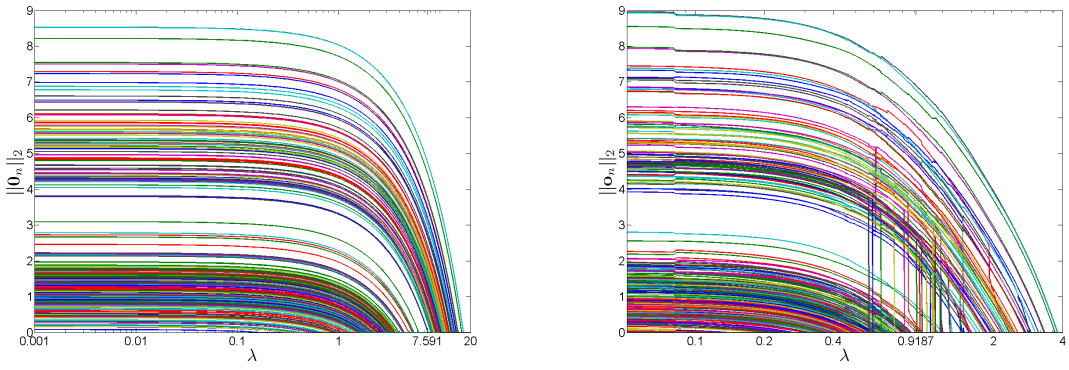
(a) Dataset with $C = 4$ spherical clusters. (b) Dataset with $C = 2$ concentric rings.

Fig. 1. Synthetic datasets: In-(out-)lier vectors are denoted by circles • (stars ★).

$(N = 280)$. The Gaussian distributions have different means and a common covariance matrix $0.8\mathbf{I}_2$. The second dataset comprises points belonging to $C = 2$ concentric rings as depicted in Fig. 1(b). The inner (outer) ring has 50 (150) points. It also contains 60 vectors lying in the areas between the rings and outside the outer ring corresponding to outliers $(N = 260)$. Clustering this second dataset is challenging even if outliers were not present due to the shape and multiscale nature of the clusters.

Starting from the dataset with the four spherical clusters, the effect of λ on the number of outliers identified is investigated. In Fig. 2, the values of $\{\|\mathbf{o}_n\|_2\}_{n=1}^N$ are plotted as a function of λ. The outlier-norm curves shown in Fig. 2(a) correspond to the RKM algorithm with $q = 1$ using a random initialization. For $\lambda > 17$, all $\mathbf{o}_n$'s are set to zero, while as λ approaches zero, more $\mathbf{o}_n$'s take nonzero values. Selecting $\lambda \in [6.2, 7.6]$ yields 80 outliers. Fig. 2(b) shows $\{\|\mathbf{o}_n\|_2\}_{n=1}^N$ as λ varies for the RPC algorithm assuming $\mathbf{\Sigma} = \sigma^2\mathbf{I}_p$. The curves for some $\mathbf{o}_n$'s exhibit a fast transition from zero.

In Fig. 3, the number of outliers identified, i.e., the number of input vectors $\mathbf{x}_n$ with corresponding $\|\mathbf{o}_n\|_2 > 0$, is plotted as a function of λ. Proper choice of λ enables both RKM and RPC to identify exactly the 80 points, which were generated as outliers in the "ground truth" model. For the RKM algorithm with $q = 1$, there is a plateau for values of $\lambda \in [6.2, 7.6]$. This plateau defines a transition region between the values of λ that identify true outliers and values of λ which erroneously deem non-outliers as outliers. Although the plateau is not present for $q > 1$, the curves show an increase in their slope for $\lambda < 5$ indicating that non-outliers are erroneously labeled as outliers. RPC with $\lambda = 0.91$ correctly identifies the 80 outlying vectors. Notice that the range of λ values for which outliers are correctly identified is smaller than the one for RKM due to the scaling of λ by σ.



(a) RKM algorithm for $q = 1$.

(b) RPC algorithm.

Fig. 2. Curves of $\|\mathbf{o}_n\|_2$'s as a function of λ for the dataset in Fig. 1(a).

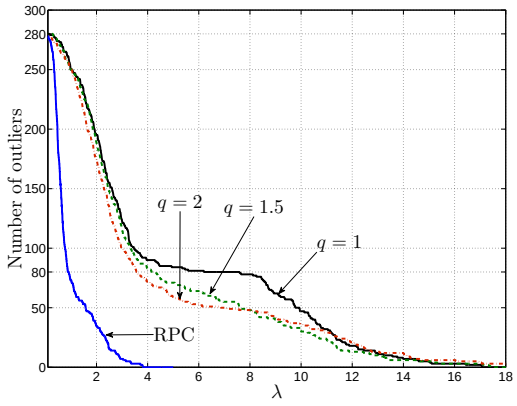


Fig. 3. Number of outliers identified as a function of λ for the dataset in Fig. 1(a).

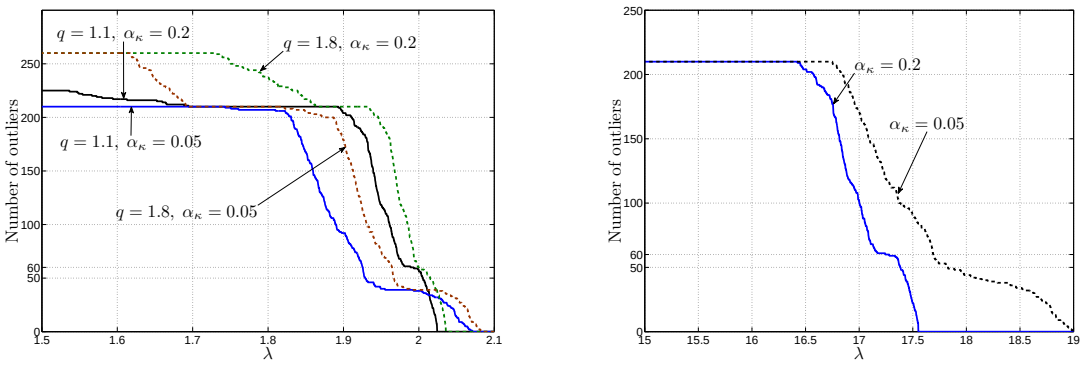
Table I shows the root-mean-squared error (RMSE) of cluster center estimates averaged over 100 algorithm initializations. Two levels of outlier contamination are considered: 40 out of $N = 240$ points (approximately 17%), and 80 out of $N = 280$ (approximately 29%). Apart from the novel algorithms, hard K-means, soft K-means with $q = 1.5$, and EM are also included. The robust versions of the hard K-means, soft K-means, and EM algorithms achieve lower RMSE with extra improvement effected by the weighted RKM (WRKM) and the weighted RPC (WRPC) algorithms.

Next, we consider clustering the second nonlinearly separable dataset using the Gaussian kernel $\kappa(\mathbf{x}_n, \mathbf{x}_m) = \exp(-\alpha_\kappa \|\mathbf{x}_n - \mathbf{x}_m\|_2^2)$, where $\alpha_\kappa > 0$ is a scaling parameter. The parameter α_κ^{-1} is chosen as a robust variance estimate of the entire dataset as described in [6]. Both KRKM and KRPC

TABLE I

RMSE OF CLUSTER CENTER ESTIMATES.

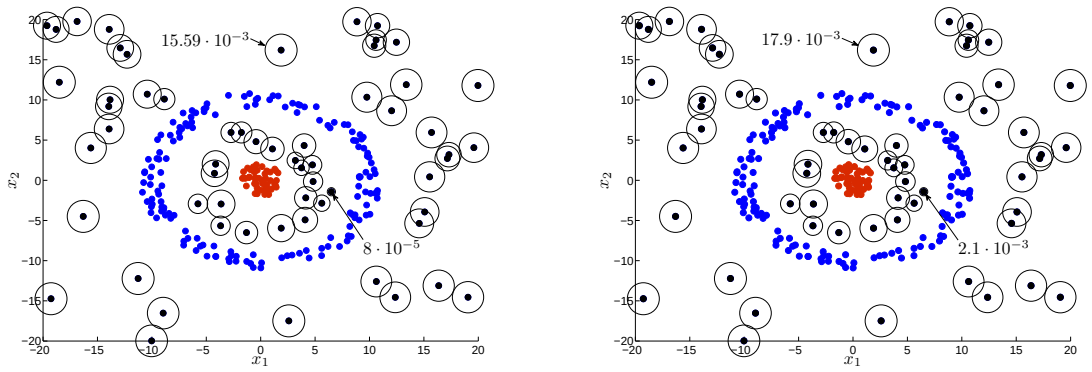
	RMSE	
Outliers/N	**40**/240	**80**/280
hard K-means	0.7227	1.0892
soft K-means	0.5530	1.0206
EM	0.6143	1.0032
hard RKM	0.4986	0.8813
hard WRKM	0.4985	0.8812
soft RKM	0.2587	0.6259
soft WRKM	0.0937	0.1758
RPC	0.2789	0.3891
WRPC	0.1525	0.1750



(a) KRKM algorithm. (b) KRPC algorithm.

Fig. 4. Number of outliers identified as a function of λ for the dataset in Fig. 1(b).

are able to identify the 60 outlying points. In Fig. 4, the number of outliers identified by KRKM and KRPC is plotted as a function of λ for different values of α_κ. Fig. 5 illustrates the values of $\|\mathbf{o}_n\|_2$'s for WKRKM and WKRPC when seeking 60 outliers. Points surrounded by a circle correspond to vectors identified as outliers, and each circle's radius is proportional to its corresponding $\|\mathbf{o}_n\|_2$ value.



(a) KRKM algorithm $(q = 1.1)$. (b) KRPC algorithm.

Fig. 5. Clustering results for the dataset in Fig. 1(b) using a Gaussian kernel with $\alpha_\kappa = 0.2$. Points surrounded by a circle were deemed as outliers; the radius of the circle is proportional to the value of $\|\mathbf{o}_n\|_2$. Smallest and largest $\|\mathbf{o}_n\|_2$ values are shown.

B. USPS Dataset

In this subsection, the robust clustering algorithms are tested on the United States Postal Service (USPS) handwritten digit recognition corpus. This corpus contains gray-scale digit images of 16×16 pixels with intensities normalized to $[-1, 1]$. It is divided to 7,201 training and 2,007 test examples of the digits 0-9. Although the corpus contains class labels, they are known to be inconsistent: some digits are erroneously labeled, while some images are difficult to be classified even by humans [25, App. A]. In this experiment, the subset of digits 0-5 is used. For each digit, both training and test sets are combined to a single set and then 300 images are sampled uniformly at random, yielding a dataset of 1800 images. Each image is represented as a 256-dimensional vector normalized to have unit ℓ_2-norm.

Hard RKM $(q = 1)$ and RPC algorithms are used to partition the dataset into $C = 6$ clusters and identify $s = 100$ outliers. All algorithms were tested for 20 Monte Carlo runs with random initializations common to all algorithms. The final partitioning is chosen as the one attaining the smallest cost in (5). The quality of the clustering is assessed through the ARI after excluding the outlier vectors. The ARI values for K-means, K-medians, and the proposed schemes are shown in Table II. Note that the ARI values for RKM (RPC) and WRKM (WRPC) are equal. This indicates that the weighted algorithms do not modify the point-to-cluster assignments already found. Interestingly, the K-medians algorithm was not able to find a partitioning of the data revealing the 6 digits present, even after 100 Monte Carlo runs.

The USPS dataset was clustered using the RKM and WRKM tuned to identify 100 outliers. WRKM is initialized with the results obtained by RKM. Although RKM and WRKM yielded the same outlier

TABLE II

ARI COEFFICIENTS FOR THE USPS DATASET ($C = 6$)

Kernel	**K-means**	**K-medians**	**RKM**	**RPC**
Linear	0.6469	0.5382	0.6573	0.6508
Polynomial	0.5571	-	0.6978	0.6965

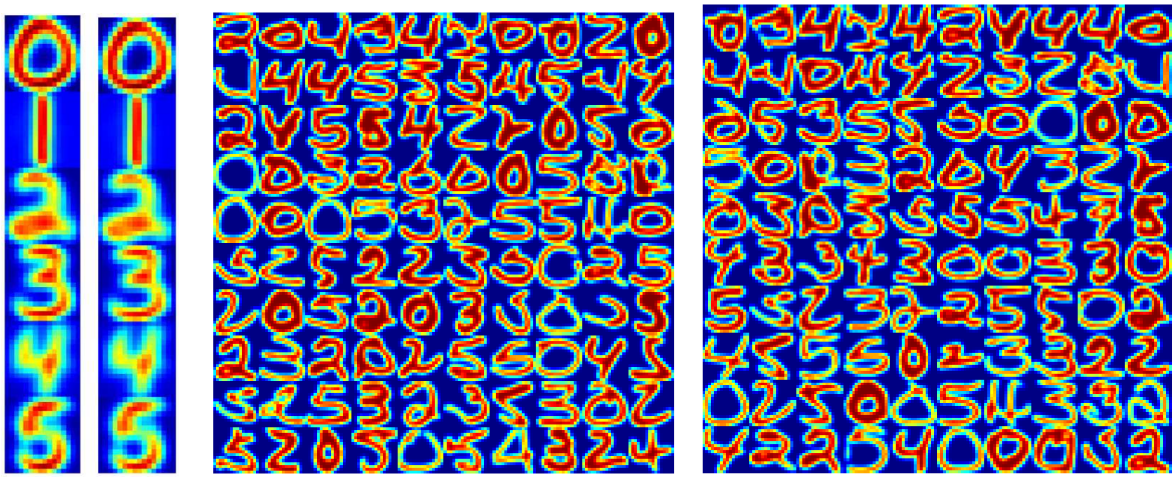
images, the size of the $\mathbf{o}_n$'s was different, becoming nearly uniform for WRKM. The USPS dataset was also clustered using the RPC and the WRPC algorithms. Fig. 6(a) shows the cluster centroids obtained by RPC and WRPC. Fig. 6(b) shows the 100 outliers identified. The outliers identified by the RPC and WRPC algorithms also coincide. The position of the outlier images in the mosaic corresponds to their ranking according to the size of their corresponding $\mathbf{o}_n$ (largest to smallest from left to right, top to bottom). Note that all outliers identified have a trait that differentiates them from the average image in each cluster. Among the 100 outliers detected by RKM and RPC, 97 are common to both approaches.

A kernelized version of the algorithms was also used on the USPS dataset. Similar to [25], the homogeneous polynomial kernel of order 3, that is $\kappa(\mathbf{x}_n, \mathbf{x}_m) = (\mathbf{x}_n^T \mathbf{x}_m)^3$, was used. The ARI scores obtained by the kernelized robust clustering algorithms are shown in Table II. Based on these scores, two important observations are in order: (i) kernelized K-means is more sensitive to outliers than K-means is; but (ii) KRKM for the particular kernel yields an improved clustering performance over RKM. Finally, the 100 outliers identified by KRKM are shown in Fig. 6(c).

C. Dolphin's Social Network

Next, KRKM is used to partition and identify outliers in a social network of $N = 62$ bottlenose dolphins living in Doubtful Sound, New Zealand [23]. Links between pairs of nodes (dolphins) represent statistically significant frequency association. The network structure is summarized by the $N \times N$ adjacency matrix $\mathbf{E}$. To identify social groups and outliers, the connection between kernel K-means and spectral clustering for graph partitioning is exploited [27]. According to this connection, the conventional spectral clustering algorithm is substituted by the kernelized K-means algorithm with a specific kernel matrix. The kernel matrix used is $\mathbf{K} = \nu \mathbf{I}_N + \mathbf{D}^{-1/2}\mathbf{E}\mathbf{D}^{-1/2}$, where $\mathbf{D} := \mathrm{diag}(\mathbf{E}\mathbf{1}_N)$ and ν is chosen larger than the minimum eigenvalue of $\mathbf{D}^{-1/2}\mathbf{E}\mathbf{D}^{-1/2}$ such that $\mathbf{K} \succ 0$.

Kernel K-means for graph partitioning is prone to being trapped at poor local minima, depending on initialization [10]. The KRKM algorithm with $C = 4$ clusters is initialized by the spectral clustering solution using the symmetric Laplacian matrix $\mathbf{L} := \mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{E}\mathbf{D}^{-1/2}$. The parameter λ is tuned to



(a) RPC and WRPC centroids. (b) Outliers identified by RPC and WRPC. (c) Outliers identified by KRKM using the polynomial kernel of order 3.

Fig. 6. Clustering and outliers for the USPS dataset with $C = 6$ tuned to identify $s = 100$ outliers.

identify $s = 12$ outliers. Fig. 7 depicts the network partitioning and the outliers identified. The results show that several nodes identified as outliers have unit degree. Having a single link indicates that nodes are marginally connected to their corresponding clusters thus deemed as outliers.

Other outlier instances adhere to more complicated structures within the network. Node *zig* has a single link, yet it is not identified as an outlier possibly due to the reduced size of its cluster, especially since four other nodes in the same cluster are identified as outliers. Interestingly, nodes *sn89*, *sn100*, *tr99* and *kringel*, with node degrees 2, 7, 7, and 9, respectively, are also identified as outliers. Using gender information about the dolphins [23], we observe that *sn89* is a female dolphin assigned to a cluster dominated by males (10 males, 3 females, and 2 unobserved). Likewise, the connectivity of *sn100* and *tr99* in the graph shows that they share many edges with female dolphins in other clusters which differentiates them from other female dolphins within the same cluster. Finally, *kringel* is a dolphin connected to 6 dolphins in other clusters and only 3 dolphins in its own cluster.

D. College Football Network

KRKM is used to partition and identify outliers in a network of $N = 115$ college football teams. The college football network represents the schedule of Division I games for the season in year 2,000 [13]. Each node corresponds to a team and a link between two teams exists if they played against each other during the season. The teams are divided into $C = 12$ conferences and each team plays games with teams

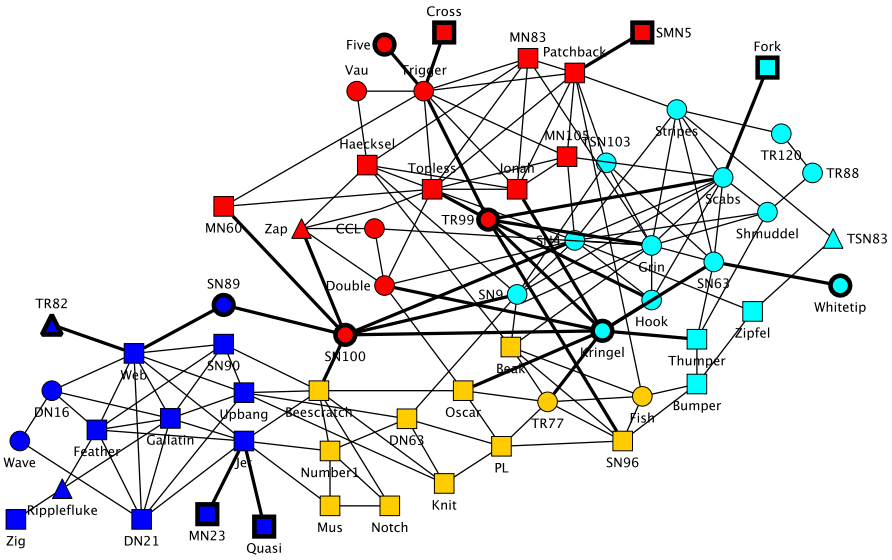


Fig. 7. KRKM clustering of the dolphin's social network: outliers are depicted bold-faced; male, female, and unobserved gender are represented by squares, circles, and triangles, respectively.

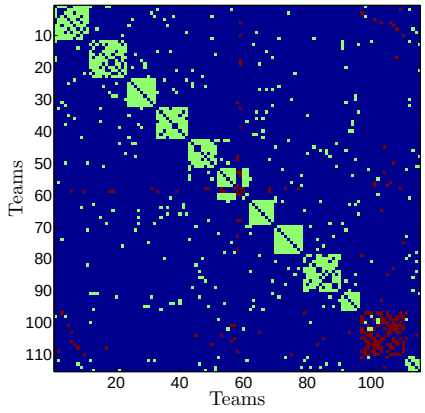


Fig. 8. The kernel matrix for the college football network permuted using KRKM clustering. Zero entries are colored blue and outliers are colored red.

in the same conference more often. KRKM is initialized via spectral clustering as described in Section V-C, while λ is tuned to identify $s = 12$ outliers. Fig. 8 shows the entries of the kernel matrix $\mathbf{K}$ after being row and column permuted so that teams in the same cluster obtained by KRKM are consecutive. The ARI coefficient yielded by KRKM after removing the outliers was 0.9218.

Teams identified as outliers sorted in descending order based on their $\|\mathbf{o}_n\|_2$ values are: Connecticut,

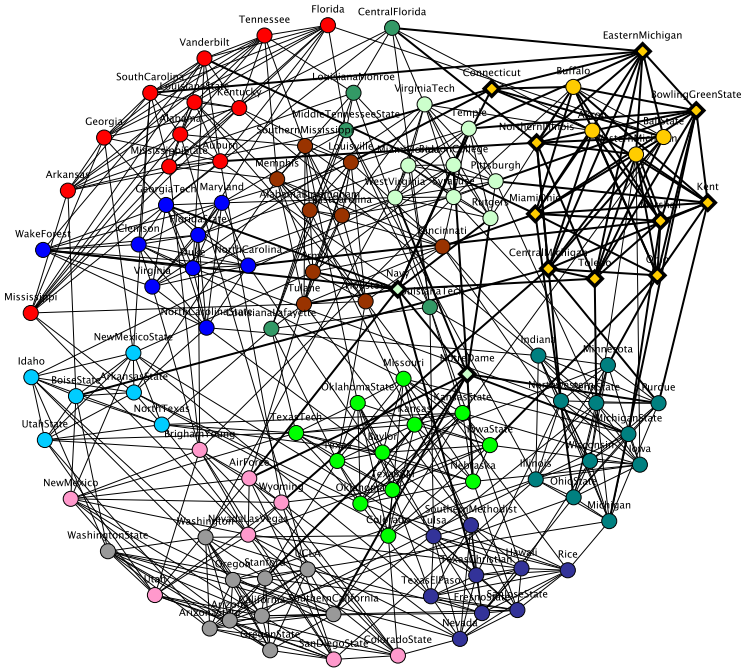


Fig. 9. Clustering of the college football network obtained by KRKM for $C = 12$. Outliers are represented by diamond-shaped nodes.

Navy, Notre Dame, Northern Illinois, Toledo, Miami (Ohio), Bowling Green State, Central Michigan, Eastern Michigan, Kent, Ohio, and Marshall. Three of them, namely Connecticut, Notre Dame, and Navy, are independent teams. Connecticut is assigned to the Mid-American conference, but it does not play as many games with teams from this conference (4 games) as other teams in the same conference do (around 8 games). Notre Dame and Navy play an equal number of games with teams from two different conferences so they could be assigned to either one. Several teams from the Mid-American conference are categorized as outliers. In hindsight, this can be explained by the subdivision of the conference into East and West Mid-American conferences. Teams in each of the Mid-American sub-conferences played about the same number of games with teams from their own sub-conference and the rest of the teams. Interestingly, using KRKM with $C = 13$ while still seeking for 12 outliers, the sub-partition of the Mid-American conference is identified. In this case, the ARI coefficient for the partition after removing outliers is 0.9110. The three independent teams, Connecticut, Notre Dame, and Navy, are again among the 12 outliers identified.

VI. Conclusions

Robust algorithms for clustering based on a principled data model accounting for outliers were developed. Both deterministic and probabilistic partitional clustering setups based on the K-means algorithm and GMM's, respectively, were considered. Exploiting the fact that outliers appear infrequently in the data, a neat connection with sparsity-aware signal processing algorithms was made. This led to the development of computationally efficient and provably convergent robust clustering algorithms. Kernelized versions of the algorithms, well-suited for high-dimensional data or when only similarity information among objects is available, were also developed. The performance of the robust clustering algorithms was validated via numerical experiments both on synthetic and real datasets.

References

[1] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA: Kluwer, 1981.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[3] L. Bobrowski and J. C. Bezdek, "C-means clustering with the l_l and l_∞ norms," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 545–554, May 1991.

[4] H.-H. Bock, "Clustering methods: A history of K-means algorithms," in *Selected Contributions in Data Analysis and Classification*, ser. Studies in Classification, Data Analysis, and Knowledge Organization, P. Brito, G. Cucumel, P. Bertrand, and F. Carvalho, Eds. Springer Berlin Heidelberg, 2007, pp. 161–172.

[5] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406 –5425, Dec. 2006.

[6] Y. Chen, X. Dang, H. Peng, and H. L. Bart, "Outlier detection with the kernelized spatial depth function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 288–305, Feb. 2009.

[7] S. Dasgupta and Y. Freund, "Random projection trees for vector quantization," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3229–3242, Jul. 2009.

[8] R. N. Davé and R. Krishnapuram, "Robust clustering methods: a unified view," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 270–293, 1997.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, pp. 1–38, Aug. 1977.

[10] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1944–1957, Nov. 2007.

[11] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pathwise coordinate optimization," *Annals of Applied Statistics*, vol. 1, no. 2, p. 302, 2007.

[12] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 450–465, May 1999.

[13] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

[15] K. Honda, A. Notsu, and H. Ichihashi, "Fuzzy PCA-guided robust k-means clustering," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 67–79, Feb. 2010.

[16] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, 2nd ed. New York: Wiley, 2009.

[17] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.

[18] J. M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with applications in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 791–802, Aug. 1991.

[19] P. Kersten, "Fuzzy order statistics and their application to fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 708–712, Dec. 1999.

[20] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, May 1993.

[21] K. Lange, D. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions (with discussion)," *J. of Comp. and Graphical Stat.*, vol. 9, pp. 1–59, 2000.

[22] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[23] D. Lusseau and M. E. J. Newman, "Identifying the role that animals play in their social networks," *Proceedings: Biological Sciences*, vol. 271, pp. S477–S481, 2004.

[24] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, Aug. 2005.

[25] B. Schölkopf and A. Smola, *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[26] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[27] I. S.Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proc. of ACM Intl. Conf. on Knowledge Discovery and Data Mining*, Seattle, WA, 2004, pp. 551–556.

[28] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 1, pp. 81–86, Jan. 1984.

[29] P. Tseng, "Convergence of block coordinate descent method for nondifferentiable minimization," *Journal on Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[30] J. Weston, A. Elisseff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *Journal of Machine Learning Research*, vol. 3, pp. 1439–1461, Mar. 2003.

[31] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[32] M. S. Yang, K. L. Wu, J. N. Hsieh, and J. Yu, "Alpha-cut implemented fuzzy clustering algorithms and switching regressions," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, pp. 588–603, 2008.

[33] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Stat. Society: Series B*, vol. 68, no. 1, pp. 49–67, Feb. 2006.

[34] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, "Gaussian mixture density modeling, decomposition, and applications," *Transactions on Image Processing*, vol. 5, no. 9, pp. 1293–1302, Sep. 1996.